Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institute for Dynamic Systems and Control
Institut für Dynamische Systeme und Regelungstechnik

Final Exam **January 24th, 2015**

Dynamic Programming & Optimal Control (151-0563-01) Prof. R. D'Andrea

Solutions

Exam Duration: **150 minutes**

Number of Problems: **5**

Permitted aids: One A4 sheet of paper.
No calculators allowed.

Problem 1 **15%**

Convert the following problems into the basic problem formulation of *Dynamic Programming.* In particular, write down the **state vector** x_k containing all relevant information at stage k, the **system dynamics** $x_{k+1} = f_k(x_k, u_k, w_k)$ describing the evolution of the state vector, the **stage cost** $g_k(x_k, u_k, w_k)$, the **final cost** $g_N(x_N)$, and the **disturbance vector** w_k with its **probability distribution** $P_k(\,\cdot\,|x_k, u_k)$. Note that in all problems the variable u_k denotes the decision variable.

a) **Problem 1:** The system dynamics are given by

$$y_{k+1} = y_k + v_k u_k + r_k u_{k-1}, \quad k = 0, 1, \ldots, N-1,$$

with $y_0 = 0$, $u_{(-1)} = 0$, and $u_k \in \mathbb{R}$. The disturbances v_k and r_k are independent random variables both taking the value 0 and 1 with equal probability. The cost function is given by

$$\text{cost} = \sum_{k=0}^{N-1} y_k^2 + u_k^2.$$

b) **Problem 2:** The system dynamics are given by

$$y_{k+1} = y_k + u_k + v_k, \quad k = 0, 1, \ldots, N-1,$$

with $y_0 = 0$, $y_{(-1)} = 0$, and $u_k \in [-1, 1]$. The disturbance v_k takes the value $1 - v_{k-1}$ and $1 + v_{k-1}$ with equal probability (we assume that $v_{(-1)} = 0$). The cost function is given by

$$\text{cost} = \sum_{k=0}^{N} y_k y_{k-1}.$$

c) **Problem 3:** The system dynamics are given by

$$y_{k+1} = y_k + u_k + v_k, \quad k = 0, 1, \ldots, N-1,$$

with $y_0 = 0$ and $u_k \in [-1, 1]$. The disturbance v_k is normally distributed with mean $a_k \in \{-1, 0, 1\}$ and variance 1. Before each stage k, an oracle tells us the value of a_k. A priori, all possible values of a_k have equal probability. The cost function is given by

$$\text{cost} = y_N^2.$$

Solution 1

In the following, the problems 1-3 are described in the basic problem formulation. Note that the solution is not unique, i.e. there are other formulations that are also correct.

a) Problem 1:

$$\text{State vector}: x_k = \begin{pmatrix} y_k \\ u_{k-1} \end{pmatrix}$$

$$\text{Dynamics}: f_k(x_k, u_k, w_k) = \begin{pmatrix} y_k + v_k u_k + r_k u_{k-1} \\ u_k \end{pmatrix}$$

$$\begin{aligned} \text{Cost}: g_k(x_k, u_k, w_k) &= y_k^2 + u_k^2 \\ g_N(x_N) &= 0 \end{aligned}$$

$$\begin{aligned} \text{Disturbance}: w_k &= \begin{pmatrix} v_k \\ r_k \end{pmatrix} \\ P_k(v_k = 0 \mid x_k, u_k) &= 0.5 \\ P_k(v_k = 1 \mid x_k, u_k) &= 0.5 \\ P_k(r_k = 0 \mid x_k, u_k) &= 0.5 \\ P_k(r_k = 1 \mid x_k, u_k) &= 0.5 \end{aligned}$$

b) Problem 2:

$$\text{State vector}: x_k = \begin{pmatrix} y_k \\ y_{k-1} \\ v_{k-1} \end{pmatrix}$$

$$\text{Dynamics}: f_k(x_k, u_k, w_k) = \begin{pmatrix} y_k + u_k + 1 + r_k v_{k-1} \\ y_k \\ 1 + r_k v_{k-1} \end{pmatrix}$$

$$\begin{aligned} \text{Cost}: g_k(x_k, u_k, w_k) &= y_k y_{k-1} \\ g_N(x_N) &= y_N y_{N-1} \end{aligned}$$

$$\begin{aligned} \text{Disturbance}: w_k &= r_k \\ P_k(r_k = -1 \mid x_k, u_k) &= 0.5 \\ P_k(r_k = 1 \mid x_k, u_k) &= 0.5 \end{aligned}$$

c) Problem 3:

$$\text{State vector}: x_k = \begin{pmatrix} y_k \\ a_k \end{pmatrix}$$

$$\text{Dynamics}: f_k(x_k, u_k, w_k) = \begin{pmatrix} y_k + u_k + v_k \\ r_k \end{pmatrix}$$

$$\begin{aligned} \text{Cost}: g_k(x_k, u_k, w_k) &= 0 \\ g_N(x_N) &= y_N^2 \end{aligned}$$

$$\begin{aligned} \text{Disturbance}: w_k &= \begin{pmatrix} v_k \\ r_k \end{pmatrix} \\ P_k(v_k \mid x_k, u_k) &= \mathcal{N}(a_k, 1) \\ P_k(r_k = -1 \mid x_k, u_k) &= 1/3 \\ P_k(r_k = 0 \mid x_k, u_k) &= 1/3 \\ P_k(r_k = 1 \mid x_k, u_k) &= 1/3 \end{aligned}$$

Problem 2 **20%**

Consider the dynamic system

$$x_{k+1} = x_k + u_k, \quad k = 0, 1,$$

with initial state $x_0 = 1$ and a discrete control input $u_k \in \{-1, 0\}$.

Note: All parts can be solved independently.

a) Assume that we want to minimize a quadratic cost function given by

$$\text{cost} = \alpha(x_2^2 + x_1^2) + (1 - \alpha)(u_1^2 + u_0^2),$$

with some known parameter $\alpha \in [0, 1]$. Apply the *Dynamic Programming Algorithm* to compute the optimal cost $J_0(x_0)$.

b) Assume that the costs cannot be combined in one single cost function. Hence, there are two cost functions to be minimized:

$$\begin{aligned} \text{cost}_1 &= x_2^2 + x_1^2, \\ \text{cost}_2 &= u_1^2 + u_0^2. \end{aligned}$$

Apply the *Dynamic Programming Algorithm* to compute the set of non-inferior solutions $\mathcal{F}_0(x_0)$.

c) Assume you found a solution to b), i.e. you computed the set of non-inferior solutions $\mathcal{F}_0(x_0)$. Explain how you can use $\mathcal{F}_0(x_0)$ to compute the solution of a), i.e. to compute the optimal cost $J_0(x_0)$, without applying the *Dynamic Programming Algorithm* again.

d) Assume you found a solution to b), i.e. you computed the set of non-inferior solutions $\mathcal{F}_0(x_0)$. Explain how you can use $\mathcal{F}_0(x_0)$ to compute the solution of a), i.e. to compute the optimal cost $J_0(x_0)$, if an additional constraint $x_2^2 + x_1^2 < 2$ has to be satisfied.

Solution 2

a) First, we write down the state space:

$$
\begin{aligned}
S_0 &= \{1\},\\
S_1 &= \{0,1\},\\
S_2 &= \{-1,0,1\}.
\end{aligned}
$$

Now, we apply the Dynamic Programming Algorithm:

- k = N = 2:

$$J_2(x_2) = \alpha x_2$$

- k = 1:

$$
\begin{aligned}
J_1(x_1) &= \min_{u_1 \in \{-1,0\}} \big\{\alpha x_1^2 + (1-\alpha)u_1^2 + J_2(x_1+u_1)\big\}\\
&= \min_{u_1 \in \{-1,0\}} \big\{\alpha x_1^2 + (1-\alpha)u_1^2 + \alpha(x_1+u_1)^2\big\}\\
&= \min_{u_1 \in \{-1,0\}} \big\{2\alpha x_1^2 + u_1^2 + 2\alpha x_1 u_1\big\}\\
J_1(0) &= \min_{u_1 \in \{-1,0\}} \big\{u_1^2\big\}\\
&= \min\big\{1,0\big\}\\
&\Rightarrow \underline{J_1(0) = 0}\\
J_1(1) &= \min_{u_1 \in \{-1,0\}} \big\{2\alpha + u_1^2 + 2\alpha u_1\big\}\\
&= \min\big\{1, 2\alpha\big\}\\
&\Rightarrow \underline{J_1(1) = \begin{cases} 2\alpha & \text{if } \alpha \le \frac{1}{2} \\ 1 & \text{if } \alpha > \frac{1}{2} \end{cases}}
\end{aligned}
$$

- k = 0:

$$
\begin{aligned}
J_0(x_0) &= \min_{u_0 \in \{-1,0\}} \big\{(1-\alpha)u_0^2 + J_1(x_0+u_0)\big\}\\
J_0(1) &= \min\big\{1-\alpha, \min\{1, 2\alpha\}\big\}\\
&= \min\big\{1-\alpha, 2\alpha\big\}\\
&\Rightarrow \underline{\underline{J_0(1) = \begin{cases} 2\alpha & \text{if } \alpha \le \frac{1}{3} \\ 1-\alpha & \text{if } \alpha > \frac{1}{3} \end{cases}}}
\end{aligned}
$$

b) First, we write down the state space:

$$
\begin{aligned}
S_0 &= \{1\},\\
S_1 &= \{0,1\},\\
S_2 &= \{-1,0,1\}.
\end{aligned}
$$

Now, we apply the Dynamic Programming Algorithm for multiobjective problems:

- k = N = 2:

$$\mathcal{F}_2(x_2) = \{(x_2^2, 0)\}$$

- k = 1:

$$\begin{aligned}
\mathcal{F}_1(x_1) &= \underset{u_1 \in \{-1,0\}}{\text{noninf}} \left\{(x_1^2 + c_1, u_1^2 + c_2) \mid (c_1, c_2) \in \mathcal{F}_2(x_1 + u_1)\right\} \\
&= \underset{u_1 \in \{-1,0\}}{\text{noninf}} \left\{(x_1^2 + x_2^2, u_1^2)\right\} \\
&= \underset{u_1 \in \{-1,0\}}{\text{noninf}} \left\{(2x_1^2 + 2x_1 u_1 + u_1^2, u_1^2)\right\} \\
\mathcal{F}_1(0) &= \underset{u_1 \in \{-1,0\}}{\text{noninf}} \left\{(u_1^2, u_1^2)\right\} \\
&= \text{noninf}\left\{(1,1),(0,0)\right\} \\
&\Rightarrow \underline{\mathcal{F}_1(0) = \left\{(0,0)\right\}} \\
\mathcal{F}_1(1) &= \underset{u_1 \in \{-1,0\}}{\text{noninf}} \left\{(2 + 2u_1 + u_1^2, u_1^2)\right\} \\
&= \text{noninf}\left\{(1,1),(2,0)\right\} \\
&\Rightarrow \underline{\mathcal{F}_1(1) = \left\{(1,1),(2,0)\right\}}
\end{aligned}$$

- k = 0:

$$\begin{aligned}
\mathcal{F}_0(x_0) &= \underset{u_0 \in \{-1,0\}}{\text{noninf}} \left\{(c_1, u_0^2 + c_2) \mid (c_1, c_2) \in \mathcal{F}_1(x_0 + u_0)\right\} \\
\mathcal{F}_0(1) &= \text{noninf}\left\{(0,1),(1,1),(2,0)\right\} \\
&\Rightarrow \underline{\underline{\mathcal{F}_0(1) = \left\{(0,1),(2,0)\right\}}}
\end{aligned}$$

c) The cost of a) is a linear combination of the two costs in b), with the weights α and $(1-\alpha)$, respectively. For each element of $\mathcal{F}_0(x_0)$, we can thus compute a scalar cost value corresponding to the cost of a). We pick the lowest one.

d) Reject all elements of $\mathcal{F}_0(x_0)$ that do not satisfy the constraint. From the remaining elements, choose the one with the lowest cost.

Problem 3 25%

The following questions are about shortest path problems. Please obey the following instructions. If you are given any output of the *Label Correcting Algorithm*, you can assume it has been generated according to these instructions.

Instructions: Recall that in the Label Correcting Algorithm only one instance of a node can be in the `OPEN` *bin at any time. If a node already in the* `OPEN` *bin enters the* `OPEN` *bin again, treat this node as if it entered the* `OPEN` *bin at the current iteration. If two nodes enter the* `OPEN` *bin in the same iteration, add the one with the lowest node number first. New nodes enter the* `OPEN` *bin from the right-hand side.*
Example: `OPEN` *bin: 2, 3, 4; Node exiting* `OPEN` *2; Nodes entering* `OPEN`*: 3, 5; new* `OPEN` *bin: 4, 3, 5.*

a) Your robot has successfully landed on Mars and just made its first picture. For reasons of robustness, the image was not stored in a single file, but in multiple chunks that overlap (see Table 1). You are really excited about the picture and want to download and publish it in tomorrow's newspaper. Unfortunately, the communication protocol with your robot only allows you to select and download one chunk at a time. Furthermore, you have a very limited bandwidth b to download the image and there is a delay d when communicating with your robot on Mars. Each chunk that you download thus takes $2d + l/b$ time, where l is the chunk length. If you download all chunks, the picture won't be ready in time for tomorrow's issue. Which chunks do you download in order to receive the complete image in minimum time?

i) Draw a graph of the shortest path problem described above and compute the arc lengths. **You don't have to solve the problem!**
Hint: Let the state be the number of consecutive bytes starting from byte 0 that you successfully downloaded.

ii) Compute a lower bound for the time required to download the remaining chunks.

Problem data:

Image size s: 2000 bytes
Number of chunks N: 8
Download bandwidth b: 10 bytes/min
Communication delay d: 10 min

start byte	end byte
0	299
200	999
300	799
400	699
600	1999
700	1699
900	1599
1500	1999

Table 1: Overlapping image data chunks.

b) Consider the shortest path problem shown in Figure 1.

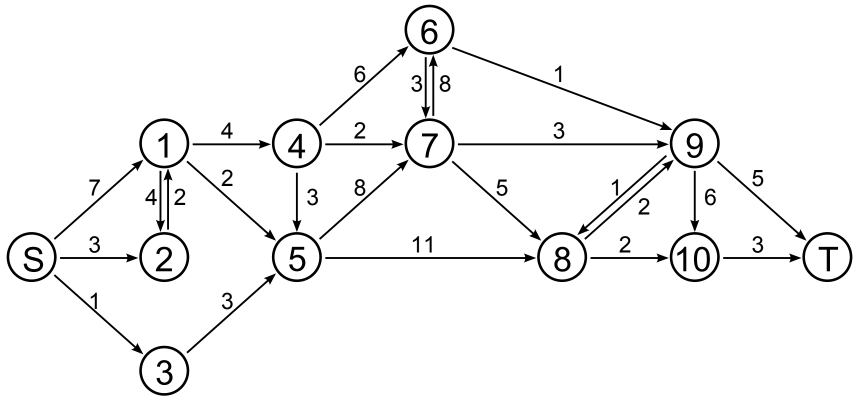

Figure 1

i) Continue table 2 for two iterations (i.e. iteration 6 and 7) using best-first search to determine at each iteration which node to remove from the `OPEN` bin.

Iteration	Node exiting `OPEN`	`OPEN`	d_S	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}	d_T
5	1	7, 8, 4	0	5	3	1	9	4	∞	12	15	∞	∞	∞
6	...													

Table 2

ii) Continue table 3 for two iterations (i.e. iteration 7 and 8) using depth-first search to determine at each iteration which node to remove from the `OPEN` bin.

Iteration	Node exiting `OPEN`	`OPEN`	d_S	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}	d_T
6	9	1, 2, 7	0	7	3	1	∞	4	∞	12	15	17	17	20
7	...													

Table 3

iii) Continue table 4 for two iterations (i.e. iteration 9 and 10) using breadth-first search to determine at each iteration which node to remove from the `OPEN` bin.

Iteration	Node exiting `OPEN`	`OPEN`	d_S	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}	d_T
8	6	4, 7, 8, 9	0	5	3	1	9	4	17	12	15	18	∞	∞
9	...													

Table 4

iv) The shortest path problem from Figure 1 can be solved using *Dynamic Programming*. Perform the initialization and one iteration step of the *Dynamic Programming Algorithm*.

c) Consider the result of the *Label Correcting Algorithm* shown in Table 5.
Note: Table 5 does not correspond to the graph in Figure 1.

Iteration	Node exiting `OPEN`	`OPEN`	d_S	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_T
0	-	S	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
1	S	1, 2	0	3	2	∞	∞	∞	∞	∞	∞	∞	∞
2	2	1, 3	0	3	2	7	∞	∞	∞	∞	∞	∞	∞
3	1	3	0	3	2	6	∞	∞	∞	∞	∞	∞	∞
4	3	4, 5, 6, 7	0	3	2	6	14	8	12	14	∞	∞	∞
5	5	4, 7, 6, 8	0	3	2	6	14	8	11	14	12	∞	∞
6	6	4, 8, 7	0	3	2	6	14	8	11	12	12	∞	17
7	7	4, 8	0	3	2	6	14	8	11	12	12	∞	17
8	8	4	0	3	2	6	14	8	11	12	12	∞	14
9	4	-	0	3	2	6	14	8	11	12	12	∞	14

Table 5

Mark the correct answer for each statement.

Grading: Each correct answer is worth 0.50%. Answers left blank are worth 0%. Wrong answers are penalized with -0.25%. The minimum score of this subproblem 3.c) is 0%.

i) Node 1 and node 2 are connected.
☐ true ☐ undetermined ☐ false

ii) The shortest path from node 6 to node T has cost
☐ 6. ☐ 3. ☐ undetermined.

iii) The cost to go from node 7 to 9 is
☐ > 5. ☐ ∞ (not connected). ☐ ≥ 5.

iv) The shortest path from node S to node 5 has cost 8.
☐ true ☐ undetermined ☐ false

v) The following method was applied in Table 5 to determine at each iteration which node exits the `OPEN` bin:
☐ breadth-first search. ☐ best-first search ☐ depth-first search.

vi) The shortest path from node S to node 4 has cost 14.
☐ true ☐ undetermined ☐ false

vii) Given a lower bound $h_j \geq 2$ to the smallest cost from j to T, $j = 1, \ldots, 9$. The number of iterations changes if you apply the A^**Algorithm* to Table 5.
☐ true ☐ undetermined ☐ false

viii) The shortest path from node 8 to node T has cost 3.
☐ true ☐ undetermined ☐ false

ix) Node 2 and node 4 are connected.
☐ true ☐ undetermined ☐ false

Solution 3

a) i) If the state is defined as the number of consecutive bytes starting from byte 0 that are already successfully downloaded, then one can move to nodes with a lower node number at zero costs. This is equivalent to throwing the overlapping bytes away. Fig. 2 shows a graph of the shortest path problem.

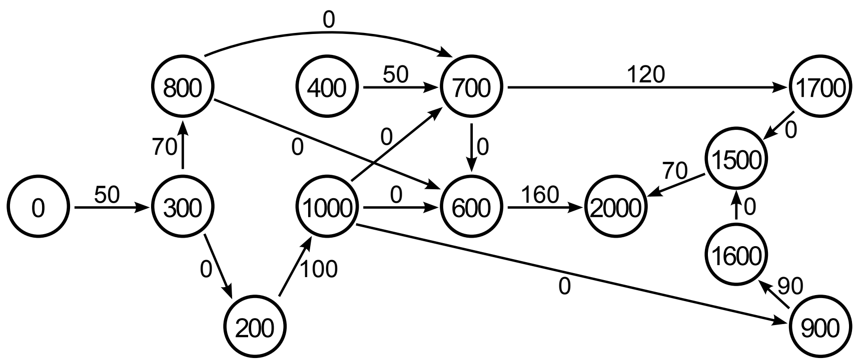

Figure 2

ii) In the best case all remaining data is stored in a single file. Starting from byte x it takes

$$\frac{s-x}{b} + 2d$$

time to download all remaining bytes.

b) i) Best-first search:

Iteration	Node exiting `OPEN`	`OPEN`	d_S	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}	d_T
5	1	7, 8, 4	0	5	3	1	9	4	∞	12	15	∞	∞	∞
6	4	8, 6, 7	0	5	3	1	9	4	15	11	15	∞	∞	∞
7	7	8, 6, 9	0	5	3	1	9	4	15	11	15	14	∞	∞

ii) Depth-first search:

Iteration	Node exiting `OPEN`	`OPEN`	d_S	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}	d_T
6	9	1, 2, 7	0	7	3	1	∞	4	∞	12	15	17	17	20
7	7	1, 2, 9	0	7	3	1	∞	4	∞	12	15	15	17	20
8	9	1, 2	0	7	3	1	∞	4	∞	12	15	15	17	20

iii) Breadth-first search:

Iteration	Node exiting `OPEN`	`OPEN`	d_S	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}	d_T
8	6	4, 7, 8, 9	0	5	3	1	9	4	17	12	15	18	∞	∞
9	4	8, 9, 6, 7	0	5	3	1	9	4	15	11	15	18	∞	∞
10	8	6, 7, 9, 10	0	5	3	1	9	4	15	11	15	17	17	∞

iv) There are $N = 11$ nodes $(S, 1, 2, \ldots, 10)$ and the terminal node T in the graph (see Fig. 1). We define the cost-to-go $J_k(i)$ to be

$$J_k(i) := \text{optimal cost of getting from } i \text{ to } t \text{ in } N - k \text{ moves.}$$

Using the *Dynamic Programming Algorithm*, the optimal cost-to-go is then computed as follows:

- $k = 10$: (Initialization)

i	$J_{10}(i)$	$\leftarrow a_{iT}$
S	∞	$a_{ST} = \infty$
1	∞	$a_{1T} = \infty$
2	∞	$a_{2T} = \infty$
3	∞	$a_{3T} = \infty$
4	∞	$a_{4T} = \infty$
5	∞	$a_{5T} = \infty$
6	∞	$a_{6T} = \infty$
7	∞	$a_{7T} = \infty$
8	∞	$a_{8T} = \infty$
9	5	$a_{9T} = 5$
10	3	$a_{10T} = 3$

- $k = 9$: (Recursion)

i	$J_9(i)$	$\leftarrow \min_{j \in \{S,1,\ldots,10\}} [a_{ij} + J_{10}(j)]$
S	∞	T cannot be reached in $\leq$ 2 steps
1	∞	T cannot be reached in $\leq$ 2 steps
2	∞	T cannot be reached in $\leq$ 2 steps
3	∞	T cannot be reached in $\leq$ 2 steps
4	∞	T cannot be reached in $\leq$ 2 steps
5	∞	T cannot be reached in $\leq$ 2 steps
6	6	$\min\{\infty, 3 + \infty, 1 + 5\}$
7	8	$\min\{\infty, 8 + \infty, 5 + \infty, 3 + 5\}$
8	5	$\min\{\infty, 2 + 5, 2 + 3\}$
9	5	$\min\{5, 1 + \infty, 6 + 3\}$
10	3	$\min\{3\}$

c) i) Undetermined.

ii) Undetermined.

iii) ≥ 5.

iv) True.

v) Best-first search.

vi) True.

vii) False.

viii) False.

ix) False.

Problem 4 **20%**

Consider the following *Stochastic Shortest Path Problem*:

$$\begin{aligned} x_{k+1} &= w_k, \\ x_k &\in \{0, 1, 2\}, \\ u_k &\in \{A, B\}. \end{aligned}$$

The transition probabilities $p_{ij}(u_k) := P(w_k = j | x_k = i, u_k)$ between the states are given by

$$\begin{array}{ll} p_{00}(A) = 1, & p_{00}(B) = 1, \\ p_{01}(A) = 0, & p_{01}(B) = 0, \\ p_{02}(A) = 0, & p_{02}(B) = 0, \\ p_{10}(A) = 0, & p_{10}(B) = 0.2, \\ p_{11}(A) = 0.2, & p_{11}(B) = 0.3, \\ p_{12}(A) = 0.8, & p_{12}(B) = 0.5, \\ p_{20}(A) = 0.5, & p_{20}(B) = 0.2, \\ p_{21}(A) = 0.3, & p_{21}(B) = 0.5, \\ p_{22}(A) = 0.2, & p_{22}(B) = 0.3. \end{array}$$

The cost function to be minimized is given by

$$\lim_{N \to \infty} \mathrm{E} \left\{ \sum_{k=0}^{N-1} g(u_k) x_k \right\},$$

with

$$g(A) = 16, \qquad g(B) = 5.$$

Note: All parts can be solved independently.

a) Explain why the state $x_k = 0$ can be considered to be the termination state of this *Stochastic Shortest Path Problem*.

b) Perform one iteration of the *Value Iteration Algorithm*, i.e. compute $J_1(1)$ and $J_1(2)$. Use $J_0(1) = 20$ and $J_0(2) = 10$ as initial guess.

c) Perform one iteration of the *Policy Iteration Algorithm*, i.e. compute $\mu^1(1)$ and $\mu^1(2)$. Use $\mu^0(1) = A$ and $\mu^0(2) = B$ as initial guess.

d) The optimal cost of the above *Stochastic Shortest Path Problem* can be solved by maximizing $J(1) + J(2)$ subject to a set of linear constraints on $J(1)$ and $J(2)$. Write down these constraints. **You don't have to simplify the constraints.**

Solution 4

a) The termination state must be cost-free. For $x_k = 0$ the stage cost is zero. Further, there must exist a policy such that from any initial state we reach the termination state with nonzero probability after a finite number of stages, and once we are in the termination state, the probability of leaving it again must be zero for all inputs. With the given transition probabilities, this is the case for $x_k = 0$.

b) We initialize the value iteration algorithm and perform one iteration:

- Initial guess:

$$\underline{J_0(1) = 20},\ \underline{J_0(2) = 10}$$

- Iteration 1:

$$\begin{aligned}
J_1(1) &= \min_{u \in \{A,B\}} \Big[g(u) \cdot 1 + p_{11}(u) J_0(1) + p_{12}(u) J_0(2) \Big] \\
&= \min \Big[16 \cdot 1 + 0.2 \cdot 20 + 0.8 \cdot 10, 5 \cdot 1 + 0.3 \cdot 20 + 0.5 \cdot 10 \Big] \\
&= \min \Big[28, 16 \Big] \quad \Rightarrow \quad \underline{\underline{J_1(1) = 16}} \\
J_1(2) &= \min_{u \in \{A,B\}} \Big[g(u) \cdot 2 + p_{21}(u) J_0(1) + p_{22}(u) J_0(2) \Big] \\
&= \min \Big[16 \cdot 2 + 0.3 \cdot 20 + 0.2 \cdot 10, 5 \cdot 2 + 0.5 \cdot 20 + 0.3 \cdot 10 \Big] \\
&= \min \Big[40, 23 \Big] \quad \Rightarrow \quad \underline{\underline{J_1(2) = 23}}
\end{aligned}$$

c) We initialize the policy iteration algorithm and perform one iteration:

- Initial guess:

$$\underline{\mu^0(1) = A},\ \underline{\mu^0(2) = B}$$

- Iteration 1:
  - Policy evaluation:

$$\begin{aligned}
J_{\mu^0}(1) &= g(A) \cdot 1 + p_{11}(A) J_{\mu^0}(1) + p_{12}(A) J_{\mu^0}(2) \\
&= 16 \cdot 1 + 0.2 \cdot J_{\mu^0}(1) + 0.8 \cdot J_{\mu^0}(2) \\
&\Rightarrow J_{\mu^0}(1) = 20 + J_{\mu^0}(2) \\
J_{\mu^0}(2) &= g(B) \cdot 2 + p_{21}(B) J_{\mu^0}(1) + p_{22}(B) J_{\mu^0}(2) \\
&= 5 \cdot 2 + 0.5 \cdot J_{\mu^0}(1) + 0.3 \cdot J_{\mu^0}(2) \\
&= 10 + 0.5 \cdot (20 + J_{\mu^0}(2)) + 0.3 \cdot J_{\mu^0}(2) \\
&= 20 + 0.8 \cdot J_{\mu^0}(2) \\
&\Rightarrow \underline{J_{\mu^0}(2) = 100} \\
&\Rightarrow \underline{J_{\mu^0}(1) = 120}
\end{aligned}$$

- Policy improvement:

$$
\begin{aligned}
\mu^1(1) &= \arg\min_{u\in\{A,B\}} \Big[g(u)\cdot 1 + p_{11}(u) J_{\mu^0}(1) + p_{12}(u) J_{\mu^0}(2) \Big] \\
&= \arg\min \Big[16\cdot 1 + 0.2\cdot 120 + 0.8\cdot 100, 5\cdot 1 + 0.3\cdot 120 + 0.5\cdot 100 \Big] \\
&= \arg\min \Big[120, 91 \Big] \quad \Rightarrow \quad \underline{\underline{\mu^1(1) = B}} \\
\mu^1(2) &= \arg\min_{u\in\{A,B\}} \Big[g(u)\cdot 2 + p_{21}(u) J_{\mu^0}(1) + p_{22}(u) J_{\mu^0}(2) \Big] \\
&= \arg\min \Big[16\cdot 2 + 0.3\cdot 120 + 0.2\cdot 100, 5\cdot 2 + 0.5\cdot 120 + 0.3\cdot 100 \Big] \\
&= \arg\min \Big[88, 100 \Big] \quad \Rightarrow \quad \underline{\underline{\mu^1(2) = A}}
\end{aligned}
$$

d) For each admissible pair $\{x, u\}$ we get one linear constraint on the optimal cost. This results in four constraints:

$$
\begin{aligned}
J(1) &\leq g(A)\cdot 1 + p_{11}(A)J(1) + p_{12}(A)J(2) \\
&= 16 + 0.2\cdot J(1) + 0.8\cdot J(2), \\
J(1) &\leq g(B)\cdot 1 + p_{11}(B)J(1) + p_{12}(B)J(2) \\
&= 5 + 0.3\cdot J(1) + 0.5\cdot J(2), \\
J(2) &\leq g(A)\cdot 2 + p_{21}(A)J(1) + p_{22}(A)J(2) \\
&= 32 + 0.3\cdot J(1) + 0.2\cdot J(2), \\
J(2) &\leq g(B)\cdot 2 + p_{21}(B)J(1) + p_{22}(B)J(2) \\
&= 10 + 0.5\cdot J(1) + 0.3\cdot J(2).
\end{aligned}
$$

Problem 5 **20%**

Consider the dynamic system

$$\dot{x}(t) = x(t)u(t)$$

with the fixed initial state $x(0) = 1$ and the fixed terminal state $x(T) = 2$. The control input $u(t)$ is constrained to $u(t) \in [u_{min}, u_{max}]$, with $u_{min} < 0 < u_{max}$.

Use the *Minimum Principle* to find the input $u^*(t)$ that minimizes the cost function

$$\int_0^T (x(t) - 1)^2 \, dt$$

for $T > \ln(2)/u_{max}$. Show that your solution is the only one that satisfies the *Minimum Principle*.

Solution 5

The Hamiltonian function is

$$H(x(t), u(t), p(t)) = (x(t) - 1)^2 + p(t)x(t)u(t).$$

Pontryagin's necessary conditions for optimality can be written as

State equation: $\dot{x}(t) = \dfrac{\partial H(x(t), u(t), p(t))}{\partial p}, \quad x(0) = x_0, \quad x(T) = x_T.$

$\Rightarrow \dot{x}(t) = x(t)u(t), \quad x(0) = 1, \quad x(T) = 2.$

Adjoint equation: $\dot{p}(t) = -\dfrac{\partial H(x(t), u(t), p(t))}{\partial x}.$

$\Rightarrow \dot{p}(t) = -p(t)u(t) - 2x(t) + 2.$

Control input: $u^*(t) = \arg\min_{u \in U} H(x(t), u, p(t)).$

$$\Rightarrow u^*(t) = \begin{cases} u_{max}, & p(t)x(t) < 0, \\ [u_{min}, u_{max}], & p(t)x(t) = 0, \\ u_{min}, & p(t)x(t) > 0. \end{cases}$$

Hamiltonian: $H(x(t), u(t), p(t)) = \text{constant}, \quad \forall t \in [0, T].$

We prove by contradiction that we start on a singular arc:

- $p(0) > 0$:
  Because $p(0)x(0) > 0$, the control input $u(t) = u_{min}$ is applied for some time interval. The evolution of the state $x(t)$ is then

  $$\dot{x}(t) = x(t)u_{min},$$
  $$\Rightarrow x(t) = e^{u_{min}t}.$$

  Inserting this result into the adjoint equation yields

  $$\dot{p}(t) = -p(t)u_{min} - 2e^{u_{min}t} + 2.$$

  A solution for $p(t)$ can be obtained using the ansatz

  $$p(t) = C_1 e^{-u_{min}t} + C_2 e^{u_{min}t} + C_3,$$
  $$\Rightarrow p(t) = C_1 e^{-u_{min}t} - \frac{1}{u_{min}} e^{u_{min}t} + \frac{2}{u_{min}}.$$

  To satisfy the initial condition $p(0) > 0$, C_1 has to be $> -\frac{1}{u_{min}}$. Since both $x(t)$ and $p(t)$ remain positive for all future, a switching in the control input $u(t)$ never appears and the final state $x(T) = 2$ cannot be reached.

- $p(0) < 0$:
  Because $p(0)x(0) < 0$, the control input $u(t) = u_{max}$ is applied for some time interval. Following the same argument as for the case $p(0) < 0$, a switching in the control input $u(t)$ never appears and $x(T) = e^{u_{max}T}$. The final constraint $x(T) = 2$ is only satisfied if $T = \frac{\ln 2}{u_{max}}$. The only possibility left for $T > \frac{\ln 2}{u_{max}}$ is to start on a singular arc.

- $p(0) = 0$:
  A singular arc is only possible if $p(t)x(t) = 0$ for a non-trivial time interval $[0, t^*]$. Because of the initial state condition $x(0) = 1$, the adjoint has to be zero, $p(t) = 0$, $\forall t \in [0, t^*]$, i.e.

$$\begin{aligned}
&\dot{p}(t) \overset{!}{=} 0, \quad \forall t \in [0, t^*], \\
&-p(t)u(t) - 2x(t) + 2 = 0, \quad \forall t \in [0, t^*], \\
&\Rightarrow x(t) = 1, \quad \forall t \in [0, t^*],
\end{aligned}$$

  and hence

$$\begin{aligned}
&\dot{x}(t) = 0, \quad \forall t \in [0, t^*], \\
&\Rightarrow u(t) = 0, \quad \forall t \in [0, t^*].
\end{aligned}$$

  Since it requires $\frac{\ln(2)}{u_{max}}$ time to move from $x(t^*) = 1$ to $x(T) = 2$, t^* is given by

$$t^* = T - \frac{\ln(2)}{u_{max}}.$$

Therefore the optimal solution is to remain on the singular arc until $t = t^*$ and then switch to $u^*(t) = u_{max}$.

- $0 \leq t < t^*$:

$$\begin{aligned}
u^*(t) &= 0, \\
x^*(t) &= 1, \\
p^*(t) &= 0.
\end{aligned}$$

- $t^* \leq t \leq T$:

$$\begin{aligned}
u^*(t) &= u_{max}, \\
x^*(t) &= e^{u_{max}(t-t^*)}, \\
p^*(t) &= -\frac{1}{u_{max}} e^{-u_{max}(t-t^*)} - \frac{1}{u_{max}} e^{u_{max}(t-t^*)} + \frac{2}{u_{max}}.
\end{aligned}$$