Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

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

Final Exam **February 6th, 2014**

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.

Use only the provided sheets for your solutions.

Problem 1 **23%**

Consider the system

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

with some initial state $x_0 \in \mathbb{R}$. The cost to be minimized is

$$\mathop{\mathrm{E}}_{w_0, w_1} \{x_2^2 + w_0 u_0^2 + w_1 u_1^2\},$$

where the disturbance w_k takes the values 0 and 1 with equal probability.

a) Assume a discrete input $u_k \in \{-1, 1\}$ for $k = 0, 1$. Find the optimal policy μ_1 and the optimal cost-to-go J_1 at stage $k = 1$ using the *Dynamic Programming Algorithm.*

b) Assume a continuous input $u_k \in \mathbb{R}$ for $k = 0, 1$. Assume that at each time step k a forecast $y_k \in \{A, B\}$ becomes available that indicates the value of the disturbance w_k. In particular, if $y_k = A$ then w_k will take the value 0, and if $y_k = B$ then w_k will take the value 1. A priori, w_k still takes the values 0 and 1 with equal probability. Augment the system to account for this additional information, and compute the optimal policy μ_0 and the optimal cost J_0 at stage $k = 0$ using the *Dynamic Programming Algorithm.*

Solution 1

a) We apply the Dynamic Programming Algorithm to find the optimal cost-to-go and the optimal policy:

- k = N = 2:

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

- k = 1:

$$\begin{aligned}
J_1(x_1) &= \min_{u_1 \in \{-1,1\}} \mathrm{E}\Big\{ w_1 u_1^2 + J_2(x_1 + u_1) \Big\} \\
&= \min_{u_1 \in \{-1,1\}} \Big[\frac{1}{2} u_1^2 + x_1^2 + 2x_1 u_1 + u_1^2 \Big] \\
&= \min_{u_1 \in \{-1,1\}} \Big[\frac{3}{2} u_1^2 + x_1^2 + 2x_1 u_1 \Big] \\
&= \min \Big[\frac{3}{2} + x_1^2 - 2x_1, \frac{3}{2} + x_1^2 + 2x_1 \Big] \\
&= \frac{3}{2} + x_1^2 + \min \Big[-2x_1, 2x_1 \Big] \\
&\Rightarrow \mu_1(x_1) = \begin{cases} 1 & \text{if } x_1 \leq 0 \\ -1 & \text{if } x_1 > 0 \end{cases} \\
&\Rightarrow J_1(x_1) = \begin{cases} x_1^2 + 2x_1 + \frac{3}{2} & \text{if } x_1 \leq 0 \\ x_1^2 - 2x_1 + \frac{3}{2} & \text{if } x_1 > 0 \end{cases}
\end{aligned}$$

b) We augment the state such that it contains the forecast:

$$\tilde{x}_k = (x_k, y_k), \quad \text{with } y_k \in \{A, B\}.$$

For a given value of y_k, we have the following probability distributions for w_k:

$$\begin{aligned}
Q_A: \quad & P(w_k = 0) = 1 \\
& P(w_k = 1) = 0 \\
Q_B: \quad & P(w_k = 0) = 0 \\
& P(w_k = 1) = 1
\end{aligned}$$

where the a priori probabilities are $p_{Q_A} = p_{Q_B} = 0.5$. The optimal policy and the optimal cost for the augmented state are computed using the Dynamic Programming Algorithm:

$$\begin{aligned}
&J_N(x_N, y_N) = g_N(x_N), \\
&J_k(x_k, y_k) = \min_{u_k \in U_k(x_k)} \mathop{\mathrm{E}}_{w_k} \left\{ g_k(x_k, u_k, w_k) + \sum_{i=1}^{m} p_i J_{k+1}(f_k(x_k, u_k, w_k), i) \Big| y_k \right\}.
\end{aligned}$$

- k = N = 2:

$$J_2(x_2, y_2) = x_2^2$$

- k = 1:

$$
\begin{aligned}
J_1(x_1, A) &= \min_{u_1 \in \mathbb{R} w_1} \mathrm{E} \Big\{ w_1 u_1^2 + p_{Q_A} J_2(x_1 + u_1, A) + p_{Q_B} J_2(x_1 + u_1, B) | A \Big\} \\
&= \min_{u_1 \in \mathbb{R}} \Big[0 \cdot u_1^2 + \frac{1}{2}(x_1 + u_1)^2 + \frac{1}{2}(x_1 + u_1)^2 \Big] \\
&\Rightarrow \underline{\mu_1(x_1, A) = -x_1}, \quad \underline{J_1(x_1, A) = 0} \\
J_1(x_1, B) &= \min_{u_1 \in \mathbb{R} w_1} \mathrm{E} \Big\{ w_1 u_1^2 + p_{Q_A} J_2(x_1 + u_1, A) + p_{Q_B} J_2(x_1 + u_1, B) | B \Big\} \\
&= \min_{u_1 \in \mathbb{R}} \Big[1 \cdot u_1^2 + \frac{1}{2}(x_1 + u_1)^2 + \frac{1}{2}(x_1 + u_1)^2 \Big] \\
&= \min_{u_1 \in \mathbb{R}} \Big[2u_1^2 + x_1^2 + 2x_1 u_1 \Big] \\
&\Rightarrow \frac{\partial [\dots]}{\partial u_1} = 4u_1 + 2x_1 = 0 \Rightarrow u_1 = -\frac{1}{2} x_1 \\
&\Rightarrow \frac{\partial^2 [\dots]}{\partial u_1^2} = 4 > 0 \Rightarrow \text{minimum is attained} \\
&\Rightarrow \underline{\mu_1(x_1, B) = -\frac{1}{2} x_1}, \quad \underline{J_1(x_1, B) = \frac{1}{2} x_1^2}
\end{aligned}
$$

- k = 0:

$$
\begin{aligned}
J_0(x_0, A) &= \min_{u_0 \in \mathbb{R} w_0} \mathrm{E} \Big\{ w_0 u_0^2 + p_{Q_A} J_1(x_0 + u_0, A) + p_{Q_B} J_1(x_0 + u_0, B) | A \Big\} \\
&= \min_{u_0 \in \mathbb{R}} \Big[0 \cdot u_0^2 + \frac{1}{2} \cdot 0 + \frac{1}{2} \cdot \frac{1}{2}(x_0 + u_0)^2 \Big] \\
&\Rightarrow \underline{\underline{\mu_0(x_0, A) = -x_0}}, \quad \underline{\underline{J_0(x_0, A) = 0}} \\
J_0(x_0, B) &= \min_{u_0 \in \mathbb{R} w_0} \mathrm{E} \Big\{ w_0 u_0^2 + p_{Q_A} J_1(x_0 + u_0, A) + p_{Q_B} J_1(x_0 + u_0, B) | B \Big\} \\
&= \min_{u_0 \in \mathbb{R}} \Big[1 \cdot u_0^2 + \frac{1}{2} \cdot 0 + \frac{1}{2} \cdot \frac{1}{2}(x_0 + u_0)^2 \Big] \\
&= \min_{u_0 \in \mathbb{R}} \Big[\frac{5}{4} u_0^2 + \frac{1}{4} x_0^2 + \frac{1}{2} x_0 u_0 \Big] \\
&\Rightarrow \frac{\partial [\dots]}{\partial u_0} = \frac{5}{2} u_0 + \frac{1}{2} x_0 = 0 \Rightarrow u_0 = -\frac{1}{5} x_0 \\
&\Rightarrow \frac{\partial^2 [\dots]}{\partial u_0^2} = \frac{5}{2} > 0 \Rightarrow \text{minimum is attained} \\
&\Rightarrow \underline{\underline{\mu_0(x_0, B) = -\frac{1}{5} x_0}}, \quad \underline{\underline{J_0(x_0, B) = \frac{1}{5} x_0^2}}
\end{aligned}
$$

Problem 2 **23%**

Consider the street map shown in Figure 1.

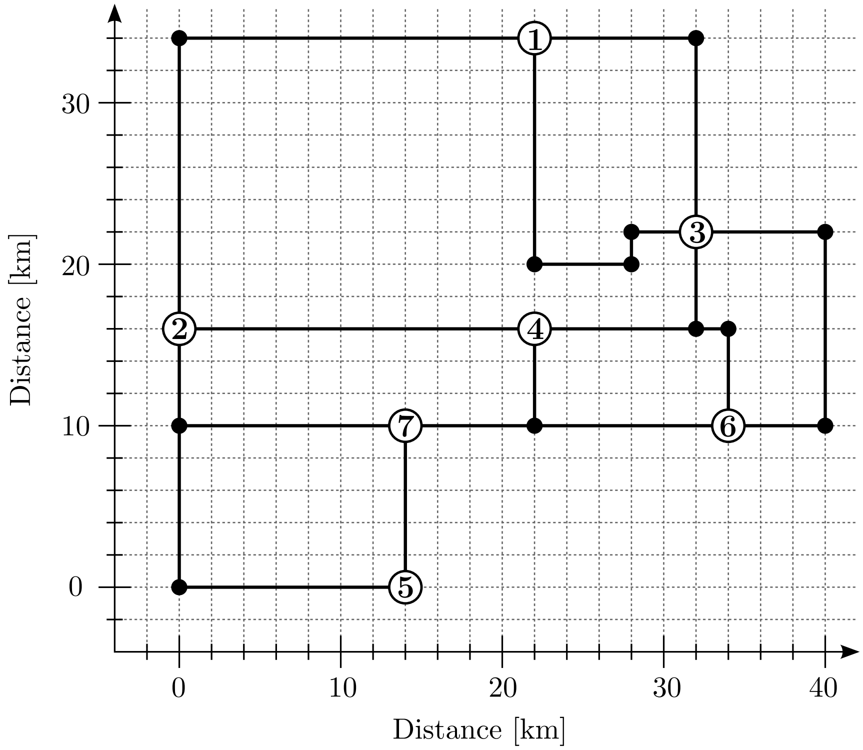

Figure 1

a) Apply the *Dynamic Programming Algorithm* to compute for each node the minimum time required to travel to the terminal node 7, given that the maximum travel speed is 1km/min.

b) Due to traffic, the true travel times are higher than computed in a) and are shown by arrows between the nodes in Figure 2 (on the back page). Calculate the minimum travel time from node 1 to 7 and the corresponding path using the A^*-*Algorithm* and the results from a) as lower bound. Use the depth-first (last-in/first-out) method to determine at each iteration which node exits the `OPEN` bin.
Instructions: Recall that only one instance of a node can be in `OPEN` *at any time. If a node that is already in the* `OPEN` *bin enters the* `OPEN` *bin again, treat this node as if it would enter the* `OPEN` *bin at the current iteration. If two nodes enter the* `OPEN` *bin in the same iteration, add the one with the largest node number first.*
Example: `OPEN` *bin: 4, 3, 2; Node exiting* `OPEN` *2 (nodes entering* `OPEN`*: 4, 5); new* `OPEN` *bin: 3, 5, 4; Node exiting* `OPEN` *4.*
Solve the problem by populating a table of the form given in Table 1. State the resulting path from node 1 to node 7 and its travel time.

Iteration	Node exiting `OPEN`	`OPEN`	d_1	d_2	d_3	d_4	d_5	d_6	d_7
0	-	...							
...									

Table 1

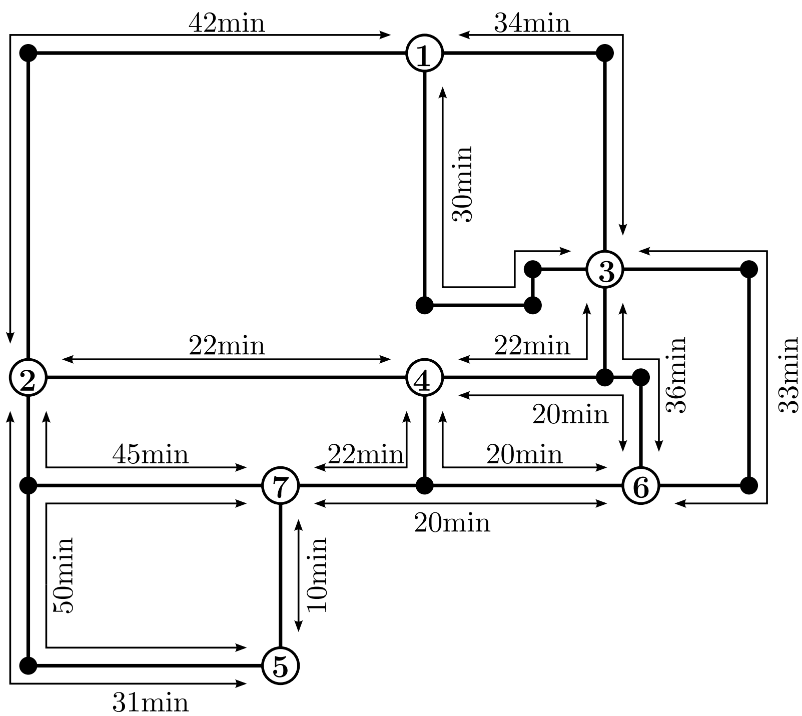
42min
34min
1
30min
3
2
22min
4
22min
36min
33min
20min
45min
22min
20min
7
6
20min
50min
10min
5
31min

Figure 2

Solution 2

a) The shortest path problem is converted to a deterministic finite-state problem. For clarity, the shortest path problem is first redrawn (see Fig. 3).

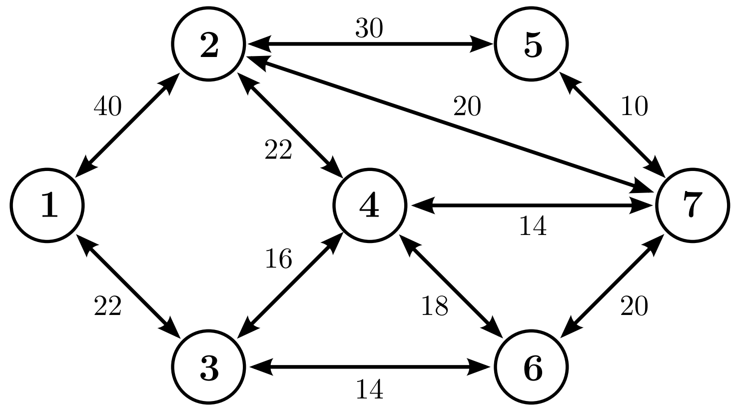

Figure 3

We have $\{1, 2, ..., N, t\}$ nodes in the graph (see Fig. 1), with $N = 6$ and the terminal node $t = 7$. 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:

- $k = 5$: (Initialization)

i	$J_5(i)$	$\leftarrow a_{it}$
1	∞	$a_{17} = \infty$
2	20	$a_{27} = 20$
3	∞	$a_{37} = \infty$
4	14	$a_{47} = 14$
5	10	$a_{57} = 10$
6	20	$a_{67} = 20$

- $k = 4$: (Recursion)

i	$J_4(i)$	$\leftarrow \min\limits_{i \in \{1,...,6\}} [a_{ij} + J_5(j)]$
1	60	$\min\{\infty, 40 + 20, 22 + \infty\}$
2	20	$\min\{20, 40 + \infty, 22 + 14, 30 + 10\}$
3	30	$\min\{\infty, 22 + \infty, 16 + 14, 14 + 20\}$
4	14	$\min\{14, 22 + 20, 16 + \infty, 18 + 20\}$
5	10	$\min\{10, 30 + 20\}$
6	20	$\min\{20, 14 + \infty, 18 + 14\}$

- $k = 3$:

i	$J_3(i)$	$\leftarrow \min_{i \in \{1,\ldots,6\}} [a_{ij} + J_4(j)]$
1	52	$\min\{60, 40+20, 22+30\}$
2	20	$\min\{20, 40+60, 22+14, 30+10\}$
3	30	$\min\{30, 22+60, 16+14, 14+20\}$
4	14	$\min\{14, 22+20, 16+30, 18+20\}$
5	10	$\min\{10, 30+20\}$
6	20	$\min\{20, 14+30, 18+14\}$

- $k = 2$:

i	$J_2(i)$	$\leftarrow \min_{i \in \{1,\ldots,6\}} [a_{ij} + J_3(j)]$
1	52	$\min\{52, 40+20, 22+30\}$
2	20	$\min\{20, 40+52, 22+14, 30+10\}$
3	30	$\min\{30, 22+52, 16+14, 14+20\}$
4	14	$\min\{14, 22+20, 16+30, 18+20\}$
5	10	$\min\{10, 30+20\}$
6	20	$\min\{20, 14+30, 18+14\}$

- $J_2(i) = J_3(i), \quad \forall i \in \{1, ..., N\} \Rightarrow$ **DONE!**

For a maximum travel speed of 1km/min, the minimum time $h(i)$ required to travel from any node i to the terminal node 7 is:

i	$h(i)$
1	52min
2	20min
3	30min
4	14min
5	10min
6	20min

b)

Iteration	Node exiting `OPEN`	`OPEN`	d_1	d_2	d_3	d_4	d_5	d_6	d_7
0	-	1	0	∞	∞	∞	∞	∞	∞
1	1	3, 2	0	42	30	∞	∞	∞	∞
2	2	3, 5, 4	0	42	30	64	73	∞	87
3	4	3, 5	0	42	30	64	73	∞[1]	86
4	5	3	0	42	30	64	73	∞	83
5	3	4	0	42	30	52	73	∞[2]	83
6	4	-	0	42	30	52	73	∞[1]	74

[1] Node 6 does not enter `OPEN` bin because $d_4 + a_{46} + h_6 > d_7$.
[2] Node 6 does not enter `OPEN` bin because $d_3 + a_{36} + h_6 > d_7$.

Minimum travel time: $\underline{\underline{d_7 = 74\text{min}}}$.

Optimal path: $\underline{\underline{1 \to 3 \to 4 \to 7}}$.

Problem 3 **23%**

Consider the following dynamic system:

$$\begin{aligned}
&x_{k+1} = w_k,\\
&x_k \in S = \{1, 2, t\},\\
&u_k \in U(x_k),\ \ U(1) = \{0.6, 1\},\ \ U(2) = \{0.5, 1\},\ \ U(t) = \{\frac{a}{2}, a\},
\end{aligned}$$

where $a \in \mathbb{R}$ is some parameter. The transition probabilities $p_{ij}(u_k) := P(w_k = j | x_k = i, u_k)$ between the states are given by

$$\begin{aligned}
p_{i1}(u_k) &= u_k q_{i1},\\
p_{i2}(u_k) &= u_k q_{i2}, \qquad\qquad \text{for } i = 1, 2, t,\\
p_{it}(u_k) &= 1 - u_k q_{i1} - u_k q_{i2}
\end{aligned}$$

with

$$\begin{aligned}
q_{11} &= 0.2, \quad q_{12} = 0.8,\\
q_{21} &= 0.4, \quad q_{22} = 0.6,\\
q_{t1} &= 0.3, \quad q_{t2} = 0.7.
\end{aligned}$$

The objective is to minimize the cost

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

with

$$g(1, u_k) = \frac{9}{u_k} - 1,\ \ g(2, u_k) = \frac{1}{u_k} + 1,\ \ g(t, u_k) = b,$$

where $b \in \mathbb{R}$ is some parameter.

Note: Part a), b), and c) can be solved independently.

a) State the range of a and b, for which the above problem can be considered to be a stochastic shortest path problem with meaningful cost. Explain your choice.

b) Assume $a = 0$ and $b = 0$. Perform one iteration of the *value iteration* algorithm, i.e compute $J_1(x_k)$. Use $J_0(1) = 0$ and $J_0(2) = 0$ as initial guess.

c) Assume $a = 0$ and $b = 0$. Use *policy iteration* to find the optimal policy $\mu^*(x_k)$ and the optimal cost $J^*(x_k)$. Use $\mu^0(1) = 1$ and $\mu^0(2) = 0.5$ as initial guess.
Hint: With this initial guess, the algorithm should converge in two or less iterations.

Solution 3

a) A stochastic shortest path problem must contain a special termination state t. Once the system reaches this termination state t it is not allowed to leave it again and no further cost should be accrued (otherwise the cost would not be finite). We get the following two conditions:

$$p_{tt}(u) = 1,\ \forall u \in U(t) \Rightarrow \underline{\underline{a = 0}},$$
$$g(t,u) = 0,\ \forall u \in U(t) \Rightarrow \underline{\underline{b = 0}}.$$

b) We initialize the value iteration algorithm and perform the first value update:

- Initial guess:

$$\underline{J_0(1) = 0},\ \underline{J_0(2) = 0}$$

- Iteration 1:

$$\begin{aligned}
J_1(1) &= \min_{u\in\{0.6,1\}} \Big[g(1,u) + p_{11}(u)J_0(1) + p_{12}(u)J_0(2)\Big] \\
&= \min\Big[9\cdot\tfrac{5}{3} - 1 + 0.6\cdot 0.2\cdot 0 + 0.6\cdot 0.8\cdot 0, 9\cdot 1 - 1 + 1\cdot 0.2\cdot 0 + 1\cdot 0.8\cdot 0\Big] \\
&= \min\Big[14, 8\Big] \quad \Rightarrow \quad \underline{\underline{J_1(1) = 8}} \\
J_1(2) &= \min_{u\in\{0.5,1\}} \Big[g(2,u) + p_{21}(u)J_0(1) + p_{22}(u)J_0(2)\Big] \\
&= \min\Big[1\cdot 2 + 1 + 0.5\cdot 0.4\cdot 0 + 0.5\cdot 0.6\cdot 0, 1\cdot 1 + 1 + 1\cdot 0.4\cdot 0 + 1\cdot 0.6\cdot 0\Big] \\
&= \min\Big[3, 2\Big] \quad \Rightarrow \quad \underline{\underline{J_1(2) = 2}}
\end{aligned}$$

c) We use the policy iteration algorithm to find the optimal policy and the corresponding optimal cost:

- Initial guess:

$$\underline{\mu^0(1) = 1},\ \underline{\mu^0(2) = 0.5}$$

- Iteration 1:

- Policy evaluation:

$$\begin{aligned}
J_{\mu^0}(1) &= g(1,1) + p_{11}(1)J_{\mu^0}(1) + p_{12}(1)J_{\mu^0}(2)\\
&= 9 \cdot 1 - 1 + 1 \cdot 0.2 \cdot J_{\mu^0}(1) + 1 \cdot 0.8 \cdot J_{\mu^0}(2)\\
&\Rightarrow J_{\mu^0}(1) = 10 + J_{\mu^0}(2)\\
J_{\mu^0}(2) &= g(2,1) + p_{21}(1)J_{\mu^0}(1) + p_{22}(1)J_{\mu^0}(2)\\
&= 1 \cdot 2 + 1 + 0.5 \cdot 0.4 \cdot J_{\mu^0}(1) + 0.5 \cdot 0.6 \cdot \cdot J_{\mu^0}(2)\\
&= 3 + 0.2 \cdot \big(10 + J_{\mu^0}(2)\big) + 0.3 \cdot J_{\mu^0}(2)\\
&= 5 + 0.5 \cdot J_{\mu^0}(2)\\
&\Rightarrow \underline{J_{\mu^0}(2) = 10}\\
&\Rightarrow \underline{J_{\mu^0}(1) = 20}
\end{aligned}$$

- Policy improvement:

$$\begin{aligned}
\mu^1(1) &= \arg\min_{u\in\{0.6,1\}} \Big[g(1,u) + p_{11}(u)J_{\mu^0}(1) + p_{12}(u)J_{\mu^0}(2)\Big]\\
&= \arg\min\Big[9 \cdot \tfrac{5}{3} - 1 + 0.6 \cdot 0.2 \cdot 20 + 0.6 \cdot 0.8 \cdot 10, 9 \cdot 1 - 1 + 1 \cdot 0.2 \cdot 20 + 1 \cdot 0.8 \cdot 10\Big]\\
&= \arg\min\Big[21.2, 20\Big] \quad \Rightarrow \quad \underline{\mu^1(1) = 1}\\
\mu^1(2) &= \arg\min_{u\in\{0.5,1\}} \Big[g(2,u) + p_{21}(u)J_{\mu^0}(1) + p_{22}(u)J_{\mu^0}(2)\Big]\\
&= \arg\min\Big[1 \cdot 2 + 1 + 0.5 \cdot 0.4 \cdot 20 + 0.5 \cdot 0.6 \cdot 10, 1 \cdot 1 + 1 + 1 \cdot 0.4 \cdot 20 + 1 \cdot 0.6 \cdot 10\Big]\\
&= \arg\min\Big[10, 16\Big] \quad \Rightarrow \quad \underline{\mu^1(2) = 0.5}
\end{aligned}$$

The algorithm has converged since $\mu^1 = \mu^0$. Hence the optimal policy and cost is:

$$\underline{\underline{\mu^*(1) = 1}},\ \underline{\underline{\mu^*(2) = 0.5}},$$

$$\underline{\underline{J^*(1) = 20}},\ \underline{\underline{J^*(2) = 10}}.$$

Problem 4 **23%**

A young farmer just took over his father's milk farm with a cow herd of size $x(0) = \frac{\alpha}{2\beta}$. He wonders whether he should increase the cow population or sell some animals. The dynamics of the cow herd can be written as

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

where the growth rate is limited to $u(t) \in [\underline{u}, \overline{u}]$. The goal of the farmer is to maximize its milk sales, which are proportional to the size of the cow herd. However, the more animals the young farmer owns, the more he has to work and the less he can enjoy life. Additionally, in order to ensure a sufficient inland milk production, the government decided to subsidize cow purchases and to penalize farmers who reduce the size of their cow population. The farmer therefore wants to find the optimal input $u^*(t)$ until his retirement T that minimizes the following cost function

$$\int_0^T -\alpha x(t) + \beta x(t)^2 - \gamma u(t) dt,$$

where $\alpha x(t)$ represents the milk sales, $\beta x(t)^2$ denotes the amount of work and $\gamma u(t)$ expresses the subsidies and penalties from the government.

Note: $\alpha, \beta, \gamma > 0$, $\underline{u} < 0 < \overline{u}$.

a) Write down the Hamiltonian function for the given problem and compute Pontryagin's necessary conditions for optimality.

b) Compute the optimal control input $u^*(t)$ of a singular arc.

c) Assume there exists an optimal solution. Show by contradiction that for $T > \sqrt{\frac{\gamma}{\beta \overline{u}}}$, the optimal control input $u^*(t)$ begins with a singular arc.
Hint: Show that the minimum principle cannot be satisfied if $p(0) > \gamma$ or $p(0) < \gamma$.

d) Assume $T > \sqrt{\frac{\gamma}{\beta \overline{u}}}$. Compute the optimal trajectories $x^*(t)$, $u^*(t)$, $t \in [0, T]$, and briefly comment the solution.

Solution 4

a) The Hamiltonian function is

$$\begin{aligned} H(x(t), u(t), p(t)) &= -\alpha x(t) + \beta x(t)^2 - \gamma u(t) + p(t)u(t), \\ &= -\alpha x(t) + \beta x(t)^2 + (p(t) - \gamma)u(t). \end{aligned}$$

Pontryagin's necessary conditions for optimality can be written as

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

$\Rightarrow \dot{x}(t) = u(t), \quad x(0) = \dfrac{\alpha}{2\beta}$

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

$\Rightarrow \dot{p}(t) = \alpha - 2\beta x(t), \quad p(T) = 0.$

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

$$\Rightarrow u^*(t) = \begin{cases} \underline{u}, & p(t) > \gamma, \\ \overline{u}, & p(t) < \gamma, \\ u(t) \in [\underline{u}, \overline{u}], & p(t) = \gamma. \end{cases}$$

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

b) A singular arc is only possible if $p(t) = \gamma$ for a non-trivial time interval $[t_1, t_2]$, $t_2 > t_1$, i.e.

$$\begin{aligned} &\dot{p}(t) \overset{!}{=} 0, \quad \forall t \in [t_1, t_2], \\ &\Rightarrow x(t) = \frac{\alpha}{2\beta}, \quad \forall t \in [t_1, t_2], \end{aligned}$$

and hence

$$\begin{aligned} &\dot{x}(t) \overset{!}{=} 0, \quad \forall t \in [t_1, t_2], \\ &\Rightarrow u(t) = 0, \quad \forall t \in [t_1, t_2]. \end{aligned}$$

The optimal control input can thus be written as

$$u^*(t) = \begin{cases} \underline{u}, & p(t) > \gamma, \\ \overline{u}, & p(t) < \gamma, \\ 0, & p(t) = \gamma \text{ and } x(t) = \frac{\alpha}{2\beta}. \end{cases}$$

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

- $p(0) > \gamma$:
  The input $u(t) = \underline{u}$ is applied for some time interval and the herd size decreases. Since $x(0) = \frac{\alpha}{2\beta}$, we know from the adjoint equation that $\dot{p}(t) > 0$, $\forall t \in (0, T]$ and hence $p(t) > \gamma$, $\forall t \in [0, T]$ which violates the final constraint on the adjoint $p(T) = 0$.

- $p(0) < \gamma$:
  The input $u(t) = \overline{u}$ is applied for some time interval and the herd size increases. Since $x(0) = \frac{\alpha}{2\beta}$, it can be derived from the adjoint equation that $\dot{p}(t) < 0$, $\forall t \in (0, T]$. Hence $p(t) < \gamma$, $\forall t \in [0, T]$ and the control input never changes, i.e. $u(t) = \overline{u}$, $\forall t \in [0, T]$. The size of the cow herd is then given by
  $$x(t) = \frac{\alpha}{2\beta} + \overline{u}t, \quad t \in [0, T],$$
  and the adjoint equation is
  $$\begin{aligned}\dot{p}(t) &= \alpha - 2\beta\left(\frac{\alpha}{2\beta} + \overline{u}t\right), \\ &= -2\beta\overline{u}t.\end{aligned}$$
  Integrating the adjoint equation once yields
  $$p(t) = c - \beta\overline{u}t^2, \quad t \in [0, T],$$
  where c is some constant that can be determined by invoking the terminal constraint.
  $$\begin{aligned}&p(T) = 0, \\ &\Rightarrow c = \beta\overline{u}T^2.\end{aligned}$$
  Given the terminal time $T > \sqrt{\frac{\gamma}{\beta\overline{u}}}$, we can conclude that $c > \gamma$ and hence $p(0) > \gamma$. However, this contradicts our initial assumption and this solution is thus not valid.

Since we assume that a solution exists, the only possibility left is that $p(0) = \gamma$ and that $u^*(t)$ begins with a singular arc.

d) We begin with a singular arc and remain on it until $t = t^*$.
$$\begin{aligned}x^*(t) &= \frac{\alpha}{2\beta}, \quad t \in [0, t^*], \\ u^*(t) &= 0, \quad t \in [0, t^*], \\ p(t) &= \gamma, \quad t \in [0, t^*].\end{aligned}$$
From the last equation, it is clear that it is not possible to remain on the singular arc the entire time because of the final constraint $p(T) = 0$. In order to decrease $p(t)$ to zero, the control input has to be switched to $u^*(t) = \overline{u}$.
$$\begin{aligned}x^*(t) &= \frac{\alpha}{2\beta} + \overline{u}(t - t^*), \quad t \in [t^*, T], \\ u^*(t) &= \overline{u}, \quad t \in [t^*, T], \\ p(t) &= \gamma - \beta\overline{u}(t - t^*)^2, \quad t \in [t^*, T].\end{aligned}$$
The switching time t^* can be determined using the final constraint on the adjoint,
$$\begin{aligned}&p(T) = \gamma - \beta\overline{u}(T - t^*)^2 \overset{!}{=} 0, \\ &\Rightarrow t^* = T - \sqrt{\frac{\gamma}{\beta\overline{u}}}.\end{aligned}$$
The optimal strategy for the farmer is therefore to keep the cow population at its initial size, which is the optimal tradeoff between work and milk production. However, when the time until the retirement is small enough, it is worth to increase the herd size and get the subsidies from the government, because after the retirement, his successor will have to deal with the additional work caused by the increased herd size.

Problem 5 8%

Consider the following Matlab script that solves a stochastic shortest path problem using *value iteration*:

```
1  % define state and input indexes
2  x1 = 1; x2 = 2;
3  u1 = 1; u2 = 2;
4
5  % define the system
6  p( :, :, u1) = [ 0.2, 0.7; 0.1, 0.9 ];
7  p( :, :, u2) = [ 0.8, 0.2; 0.7, 0.1 ];
8  g( :, u1 ) = [ 1, 1 ];
9  g( :, u2 ) = [ 2, 2 ];
10
11 % define value iteration function
12 val_it_fun = @( J_k ) min( ...
13   [ g( x1, u1 ) + p( x1, x1, u1 ) * J_k( x1 ) + p( x1, x2, u1 ) * J_k( x2 ), ...
14     g( x1, u2 ) + p( x1, x1, u2 ) * J_k( x1 ) + p( x1, x2, u2 ) * J_k( x2 ); ...
15     g( x2, u1 ) + p( x2, x1, u1 ) * J_k( x1 ) + p( x2, x2, u1 ) * J_k( x2 ), ...
16     g( x2, u2 ) + p( x2, x1, u2 ) * J_k( x1 ) + p( x2, x2, u2 ) * J_k( x2 ) ]' );
17
18 % perform value iteration until cost has converged
19 J_k = [ 0, 0 ];
20 tol = 1e-5;
21 while true
22     [ J_k( end + 1, : ), ind ] = val_it_fun( J_k( end, : ) );
23     if max( abs( J_k( end, : ) - J_k( end - 1, : ) ) ) < tol, break; end
24 end
25 J_opt = ...
26 u_opt = ...
27 number_of_iterations = ...
28
29 % display results
30 disp( [ 'terminated after ', num2str( number_of_iterations ), ' iterations' ] );
31 disp( [ 'optimal cost:  J( x1 ) = ', num2str( J_opt( x1 ) ) ] );
32 disp( [ '               J( x2 ) = ', num2str( J_opt( x2 ) ) ] );
33 disp( [ 'optimal input: u( x1 ) = ', num2str( u_opt( x1 ) ) ] );
34 disp( [ '               u( x2 ) = ', num2str( u_opt( x2 ) ) ] );
```

a) What is the size of the matrix p, i.e. what is the output if you type **size**(p) into the Matlab Command Window? And what does a particular entry p (...) of the matrix p represent?

b) The line 23 implements the stopping criteria of the algorithm. Explain in words under which condition the iteration loop is exited.

c) Complete the lines 25, 26, and 27, such that the script displays the optimal cost, the indexes of the optimal inputs, and the number of value iterations that were performed.

Solution 5

a) The matrix p has the size $2 \times 2 \times 2$, i.e. typing **size**(p) into the Matlab Command Window returns [2, 2, 2]. A particular entry p(i, j, u) represents the transition probability from state i to state j if the control input u is applied.

b) The loop is exited if for all states the absolute value of the cost difference to the previous iteration is below a threshold.

c) The optimal cost, the indexes of the optimal inputs, and the number of iterations is obtained by the following lines of code:

```
J_opt = J_k( end, : );
u_opt = ind;
number_of_iterations = size( J_k, 1 ) - 1;
```