ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

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

Final Exam **January 22, 2013**

Dynamic Programming & Optimal Control (151-0563-00) Angela Schoellig

Solutions

Exam Duration: **150 minutes**

Number of Problems: **4**

Permitted aids: One A4 sheet of paper.

Use only the provided sheets for your solutions.

Problem 1 25%

Consider the following discrete-time system

$$x_{k+1} = x_k + u_k + w_k, \quad x_k, u_k, w_k \in \mathbb{R}, \quad k = 0, 1,$$

where x_k is the state of the system at stage k, u_k is the input and w_k is a (piecewise) continuous disturbance uniformly distributed between -1 and $+1$. Finally, the cost function to be minimized is given by

$$\underset{w_0, w_1}{\mathrm{E}} \left\{ x_2^2 + u_0^2 + u_1^2 \right\}.$$

a) Explain what the cost $\underset{w_0, w_1}{\mathrm{E}} \left\{ x_2^2 + u_0^2 + u_1^2 \right\}$ penalizes.

b) Find the optimal policy $u_1^* = \mu_1(x_1)$ and the optimal cost-to-go $J_1(x_1)$[1].

c) Assume $w_k = 0$ for all k, i.e. there is no disturbance. Find the optimal policy $u_1^* = \mu_1(x_1)$ and the optimal cost-to-go $J_1(x_1)$.

d) Compare the optimal cost-to-go in **b)** and **c)** and give an intuitive explanation for your observation.

e) How does the optimal policy change in **c)** if the input is constrained by $0 \leq u_k \leq 1$ for all k ?

[1] Hint: Let x be a continuous random variable with probability density function $p(x)$. The expected value of an arbitrary function $g(x)$ is given by

$$\mathrm{E}\left\{g(x)\right\} = \int_{-\infty}^{+\infty} g(x)p(x)dx.$$

Solution 1

a) The cost penalizes the control effort and the final state error, i.e. the deviation of the final state from zero; the solution will be a trade-off between minimizing the input and minimizing the final state error.

b) The optimal control problem is considered over a time horizon $N = 2$ and the cost to be minimized is defined by

$$g_2(x_2) = x_2^2 \quad \text{and} \quad g_k(x_k, u_k, w_k) = u_k^2,\ k = 0, 1.$$

We apply the Dynamic Programming Algorithm:
Stage 2:

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

Stage 1:

$$\begin{aligned} J_1(x_1) &= \min_{u_1} \left[\mathrm{E}_{w_1} \left\{ u_1^2 + J_2(x_2) \right\} \right] \\ &= \min_{u_1} \left[\mathrm{E}_{w_1} \left\{ u_1^2 + (x_1 + u_1 + w_1)^2 \right\} \right] \\ &= \min_{u_1} \left[\mathrm{E}_{w_1} \left\{ u_1^2 + (x_1 + u_1)^2 + w_1^2 + (x_1 + u_1) w_1 \right\} \right] \\ &= \min_{u_1} \left[u_1^2 + (x_1 + u_1)^2 + \mathrm{E}_{w_1}\{w_1^2\} + (x_1 + u_1) \mathrm{E}_{w_1}\{w_1\} \right] \end{aligned}$$

Now, using $\mathrm{E}_{w_1}\{w_1\} = 0$ and $\mathrm{E}_{w_1}\{w_1^2\} = \int_{-1}^{+1} w_1^2 p(w_1) dw_1 = \frac{1}{2} \int_{-1}^{+1} w_1^2 dw_1 = \frac{1}{3}$,

$$J_1(x_1) = \min_{u_1} \left[u_1^2 + (x_1 + u_1)^2 + \frac{1}{3} \right] =: \min_{u_1} C_1(u_1).$$

The minimum is attained at a u_1 for which the gradient with respect to u_1 is zero; that is,

$$\frac{\partial C_1}{\partial u_1} = 2u_1 + 2(x_1 + u_1) = 0 \quad \Rightarrow \quad u_1 = -\frac{1}{2} x_1.$$

Since the second derivative $\frac{\partial^2 C_1}{\partial u_1^2} = 4 > 0$, $u_1 = -\frac{1}{2} x_1$ is the minimizing input.
Finally, the corresponding optimal cost-to-go is given by

$$J_1^*(x_1) = \frac{1}{2} x_1^2 + \frac{1}{3}.$$

c) Similar calculations as in the above part yield

$$u_1^* = -\frac{1}{2} x_1 \quad \text{and} \quad J_1^*(x_1) = \frac{1}{2} x_1^2.$$

d) Although the optimal policies are the same, due to the quadratic term x_2^2 in the cost and the linear term of the disturbance in the dynamics with coefficient 1, the expected value of the optimal cost-to-go in the presence of a disturbance is increased by the variance of the disturbance w_1, $\mathrm{E}_{w_1}\{w_1^2\} = \frac{1}{3}$, compared to the case where there is no disturbance.

e) We obtain:

1. if $x_1 \leq -2$: $u_1^* = 1$,
2. if $-2 < x_1 \leq 0$: $u_1^* = -\frac{1}{2} x_1$,
3. if $x_1 > 0$: $u_1^* = 0$.

Problem 2 25%

Consider the transition graph shown in Figure 1.

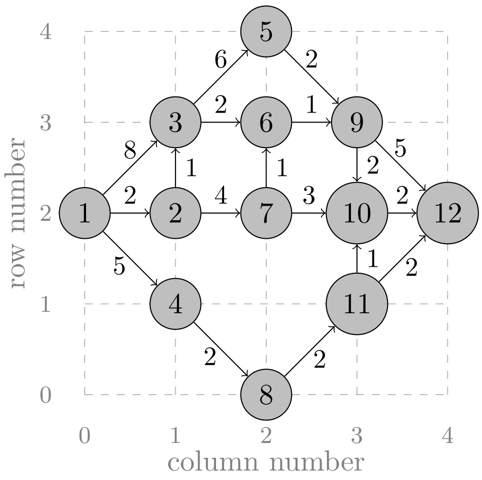


Figure 1: Transition graph of the shortest path problem.

a) Calculate the shortest path from node 1 to node 12 and the corresponding optimal cost using the Label Correcting Algorithm. Use the depth-first (last-in/first-out) method to determine at each iteration which node exits the OPEN bin.
Instructions: If a node that is already in the OPEN *bin enters the* OPEN *bin again, remove the one that has already been in the* OPEN *bin. If two nodes enter the* OPEN *bin in the same iteration, add the one with the largest node number first. Example:* OPEN *bin: 2, 3, 4; Node exiting* OPEN*: 2 (nodes entering* OPEN*: 3, 7); new* OPEN *bin: 3, 7, 4; Node exiting* OPEN*: 3.*
From a secret source you know that the distance from 1 to 12 is smaller than 13. Make use of this information.
Solve the problem by populating a table of the following form[2]: State the resulting shortest

Iteration	Node exiting OPEN	OPEN	d_1	d_2	d_3	...	d_{12}
0	-	...					
1	1	...					

path and its associated cost.

b) Assume all nodes are on a grid as indicated in Figure 1. The problem is again to find the shortest path from node 1 to node 12. The distance d_{ij} between two nodes i and j satisfies the following equation:

$$d_{ij} \geq |r_i - r_j| + |c_i - c_j| \, ,$$

where r_i is the number of the row of node i and c_i is the number of the column of node i. Use this information to strengthen the condition on whether a node enters the OPEN bin of the algorithm that was given in **a)**. Solve the problem by populating a similar table.

c) Can this problem also be solved using the backwards Dynamic Programming Algorithm? Give a short explanation.

d) Can this problem also be solved using the forward Dynamic Programming Algorithm? Give a short explanation.

[2] Please use the paper in landscape orientation for the table.

Solution 2

a)

Iter.	Node exiting `OPEN`	`OPEN`	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}	d_{11}	d_{12}
0	-	1	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	13
1	1	2, 3, 4	0	2	8	5	∞	∞	∞	∞	∞	∞	∞	13
2	2	3, 7, 4	0	2	3	5	∞	∞	6	∞	∞	∞	∞	13
3	3	5, 6, 7, 4	0	2	3	5	9	5	6	∞	∞	∞	∞	13
4	5	9, 6, 7, 4	0	2	3	5	9	5	6	∞	11	∞	∞	13
5	9	6, 7, 4	0	2	3	5	9	5	6	∞	11	∞	∞	13
6	6	9, 7, 4	0	2	3	5	9	5	6	∞	6	∞	∞	13
7	9	10, 7, 4	0	2	3	5	9	5	6	∞	6	8	∞	11
8	10	7, 4	0	2	3	5	9	5	6	∞	6	8	∞	10
9	7	4	0	2	3	5	9	5	6	∞	6	8	∞	10
10	4	8	0	2	3	5	9	5	6	7	6	8	∞	10
11	8	11	0	2	3	5	9	5	6	7	6	8	9	10
12	11	-	0	2	3	5	9	5	6	7	6	8	9	10

Optimal path: $1 \mapsto 2 \mapsto 3 \mapsto 6 \mapsto 9 \mapsto 10 \mapsto 12$ Cost: 10

b)

Iter.	Node exiting `OPEN`	`OPEN`	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}	d_{11}	d_{12}
0	-	1	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	13
1	1	2, 3, 4	0	2	8	5	∞	∞	∞	∞	∞	∞	∞	13
2	2	3, 7, 4	0	2	3	5	∞	∞	6	∞	∞	∞	∞	13
3	3	6, 7, 4	0	2	3	5	∞	5	6	∞	∞	∞	∞	13
4	6	9, 7, 4	0	2	3	5	∞	5	6	∞	6	∞	∞	13
5	9	10, 7, 4	0	2	3	5	∞	5	6	∞	6	8	∞	11
6	10	7, 4	0	2	3	5	∞	5	6	∞	6	8	∞	10
7	7	4	0	2	3	5	∞	5	6	∞	6	8	∞	10
8	4	-	0	2	3	5	∞	5	6	∞	6	8	∞	10

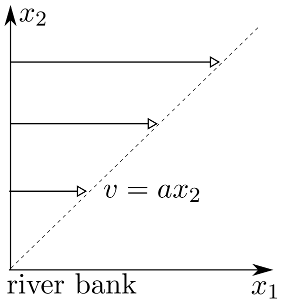
Optimal path: $1 \mapsto 2 \mapsto 3 \mapsto 6 \mapsto 9 \mapsto 10 \mapsto 12$ Cost: 10

c) Yes, a shortest path problem can be converted to a deterministic Dynamic Programming problem.

d) Yes, since the optimal control problem is deterministic.

Problem 3 25%

Consider a sufficiently wide and straight river, where the water speed v linearly increases with the distance x_2 from the river bank, i.e. $v(x_2) = ax_2$, where $a > 0$, $x_2 > 0$.



The dynamics of a boat on the river is given by

$$\begin{aligned}
\dot{x}_1(t) &= ax_2(t) + u_1(t), \\
\dot{x}_2(t) &= u_2(t), \qquad t \in [0, T], \quad x_1(0) = 0,\ x_2(0) = 0,
\end{aligned}$$

where x_1 is the position of the boat along the straight river bank, x_2 is the distance from the river bank, u_1, u_2 are the inputs along x_1 and x_2, and T is fixed.
The control objective is to maximize $x_1(T)$, the distance travelled along the river bank. Note that the objective does not require $x_2(T)$ to be zero.

a) Find the optimal input $(u_1^*(t), u_2^*(t))$ for $t \in [0, T]$ under the constraints $|u_1| \leq 1$ and $|u_2| \leq 1$.

b) Find the optimal input $(u_1^*(t), u_2^*(t))$ for $t \in [0, T]$ under the constraint $u_1^2 + u_2^2 \leq 1$.

Solution 3

The boundary conditions of the continuous-time optimal control problem are $x_1(0) = 0$ and $x_2(0) = 0$. The objective is to minimize $-x_1(T)$. Let $x := (x_1, x_2)^{\mathrm{T}}$, $u := (u_1, u_2)^{\mathrm{T}}$ and $p := (p_1, p_2)^{\mathrm{T}}$.

Applying the Minimum Principle:

With stage cost $g(x, u) = 0$ and terminal cost $h(x) = -x_1(T)$ the Hamiltonian is given by

$$\begin{aligned} H(x,u,p) &= 0 + \begin{bmatrix} p_1 & p_2 \end{bmatrix} \begin{bmatrix} ax_2 + u_1 \\ u_2 \end{bmatrix} \\ &= p_1(ax_2 + u_1) + p_2 u_2. \end{aligned}$$

The adjoint equations

$$\begin{aligned} \dot{p}_1(t) &= -\frac{\partial H}{\partial x_1} = 0, \quad \Rightarrow \quad p_1(t) = \text{constant for } t \in [0,T], \\ \dot{p}_2(t) &= -\frac{\partial H}{\partial x_2} = -ap_1(t) \end{aligned}$$

with the boundary condition

$$p_1(T) = \frac{\partial h}{\partial x_1} = -1 \quad \text{and} \quad p_2(T) = \frac{\partial h}{\partial x_2} = 0$$

result in

$$p_1(t) = -1 \quad \text{and} \qquad p_2(t) = a(t-T).$$

The optimal input $(u_1^*(t), u_2^*(t))$ is obtained by minimizing the Hamiltonian along the optimal trajectory

$$\begin{aligned} u^*(t) &= \underset{(u_1,u_2)}{\operatorname{argmin}} \left[-(ax_2 + u_1) + a(t-T)u_2 \right] \\ &= \underset{(u_1,u_2)}{\operatorname{argmin}} \left[-u_1 - a(T-t)u_2 \right] =: \underset{(u_1,u_2)}{\operatorname{argmin}} \left[C_t \right], \quad t \in [0,T]. \end{aligned} \tag{1}$$

a) $|u_1| \leq 1$ and $|u_2| \leq 1$:
Since the Hamiltonian is linear in u_1, u_2 with negative coefficients, the minimizing input is given by $(u_1^*(t), u_2^*(t)) = (1, 1)$, $t \in [0, T]$.

b) $u_1^2 + u_2^2 \leq 1$:
Method 1: Since the Hamiltonian is linear in u_1, u_2 and the constraint set is convex the optimal solution will lie on the border, i.e. $u_1 = \sqrt{1 - u_2^2}$. Note that $u_1 = -\sqrt{1 - u_2^2}$ is not an optimal candidate due to the negative coefficient of u_1 in the Hamiltonian, which needs to be minimized.

This implies,

$$u_2^*(t) = \underset{u_2}{\operatorname{argmin}} \left[-\sqrt{1 - u_2^2} - a(T-t)u_2 \right] =: \underset{u_2}{\operatorname{argmin}} \left[\bar{C}_t(u_2) \right]. \tag{2}$$

The minimum is attained at a u_2 for which the gradient with respect to u_2 is zero; that is,

$$\frac{\partial \bar{C}_t}{\partial u_2} = \frac{u_2}{\sqrt{1 - u_2^2}} - a(T-t) = 0 \quad \Rightarrow \quad u_2 = \frac{a(T-t)}{\sqrt{1 + a^2(T-t)^2}}.$$

Since the second derivative

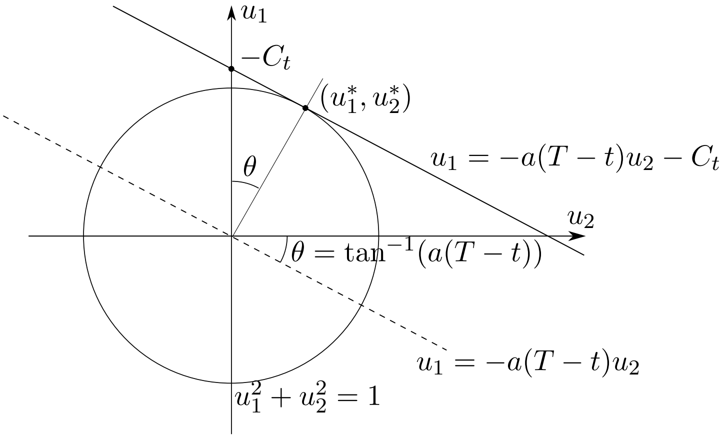
$$\left.\frac{\partial^2 \bar{C}_t}{\partial u_2^2}\right|_{u_2=\frac{a(T-t)}{\sqrt{1+a^2(T-t)^2}}} = (1+a^2(T-t)^2)^{\frac{3}{2}} > 0,$$

$u_2 = \frac{a(T-t)}{\sqrt{1+a^2(T-t)^2}}$ is the minimizing input.

Note that here again we exclude $u_2 = -\frac{a(T-t)}{\sqrt{1+a^2(T-t)^2}}$ from the optimal candidate due to the negative coefficient of u_2 in the Hamiltonian, which needs to be minimized. Finally the optimal input is given by

$$(u_1^*(t), u_2^*(t)) = (\frac{1}{\sqrt{1+a^2(T-t)^2}}, \frac{a(T-t)}{\sqrt{1+a^2(T-t)^2}})$$

Method 2:(graphical)



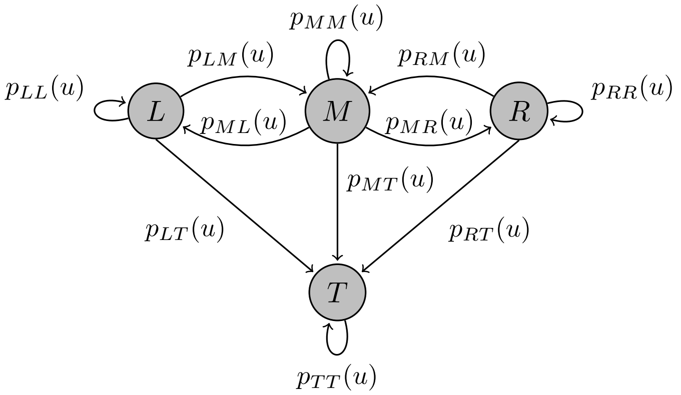
Graphically, minimizing C_t translates into maximizing the intercept $-C_t$ of the line $u_1 = -a(T-t)u_2 - C_t$. This yields

$$(u_1^*(t), u_2^*(t)) = (\frac{1}{\sqrt{1+a^2(T-t)^2}}, \frac{a(T-t)}{\sqrt{1+a^2(T-t)^2}}).$$

Problem 4 25%

Mr. Lucky's goal in life is to win the jackpot in a lottery and become rich. The three cities Lefttown (L), Middletown (M), and Righttown (R) offer daily lotteries. Let $x_k \in \{L, M, R\}$ is the location of Mr. Lucky on the kth day. However, he is only allowed to participate in the lottery of the city in which he is currently staying. Each day he wins either a ticket to go to another city, a hotel voucher to stay in his current city, or the jackpot. He is always making use of his prize on the same day. If he wins the jackpot, he retires and lives happily ever after ($x_k = T$).

The above stochastic shortest path problem is represented by the following transition graph:



Every city's lottery is offering different plans from which Mr. Lucky can choose each day (e.g. if he is in city L he can choose lottery plan a_L or b_L),

$$\begin{aligned} U(L) &= \{a_L, b_L\} \\ U(M) &= \{a_M, b_M, c_M\} \\ U(R) &= \{a_R, b_R\} \\ U(T) &= \{a_T\}, \end{aligned}$$

resulting in different probabilities for winning the prizes:

$p_{LL}(a_L) = 1/2$	$p_{ML}(a_M) = 1/4$	$p_{MR}(b_M) = 0$	$p_{RR}(a_R) = 1/2$	$p_{TT}(a_T) = 1.$
$p_{LM}(a_L) = 1/4$	$p_{MM}(a_M) = 0$	$p_{MT}(b_M) = 3/4$	$p_{RM}(a_R) = 1/4$	
$p_{LT}(a_L) = 1/4$	$p_{MR}(a_M) = 1/4$	$p_{ML}(c_M) = 1/4$	$p_{RT}(a_R) = 1/4$	
$p_{LL}(b_L) = 1/2$	$p_{MT}(a_M) = 1/2$	$p_{MM}(c_M) = 1/4$	$p_{RR}(b_R) = 0$	
$p_{LM}(b_L) = 1/2$	$p_{ML}(b_M) = 0$	$p_{MR}(c_M) = 1/4$	$p_{RM}(b_R) = 1/2$	
$p_{LT}(b_L) = 0$	$p_{MM}(b_M) = 1/4$	$p_{MT}(c_M) = 1/4$	$p_{RT}(b_R) = 1/2$	

The objective is to find a stationary policy $\mu(x_k)$ for Mr. Lucky that minimizes the expected time to win the jackpot starting from a given initial city $i \in \{L, M, R\}$, i.e. a policy that minimizes the following cost

$$J_\pi(i) = \lim_{N\to\infty} \mathrm{E}\left\{ \sum_{k=0}^{N-1} g\left(x_k, \mu\left(x_k\right)\right) | x_0 = i \right\} \text{ with } g(x_k, \mu(x_k)) = \begin{cases} 0 \text{ if } x_k = T, \\ 1 \text{ otherwise.} \end{cases}$$

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

a) Find the optimal policy $\mu(x_k)$, $x_k \in \{L, M, R\}$ for the given problem using policy iteration. Start with evaluating the initial policies $\mu^0(L) = a_L$, $\mu^0(M) = c_M$, and $\mu^0(R) = a_R$.

b) Using value iteration, perform two iterations for the given problem and state the minimizing inputs at each iteration. Start with the initial values $J^0(L) = J^0(M) = J^0(R) = 4$.

c) How do you know when to terminate the value iteration algorithm?

Solution 4

a) **Iteration 1:**

Policy evaluation:

$$\begin{aligned}
J^1(L) &= 1 + p_{LL}(a_L)J^1(L) + p_{LM}(a_L)J^1(M) + p_{LT}(a_L)\underbrace{J^1(T)}_{=0} \\
&= 1 + \frac{1}{2}J^1(L) + \frac{1}{4}J^1(M) \\
&\Rightarrow J^1(L) = 2 + \frac{1}{2}J^1(M) \qquad (3) \\
J^1(R) &= 1 + p_{RR}(a_R)J^1(R) + p_{RM}(a_R)J^1(M) + p_{RT}(a_R)\underbrace{J^1(T)}_{=0} \\
&= 1 + \frac{1}{2}J^1(R) + \frac{1}{4}J^1(M) \\
&\Rightarrow J^1(R) = 2 + \frac{1}{2}J^1(M) \qquad (4) \\
J^1(M) &= 1 + p_{ML}(c_M)J^1(L) + p_{MM}(c_M)J^1(M) + p_{MR}(c_M)J^1(R) + p_{MT}(c_M)\underbrace{J^1(T)}_{=0} \\
&= 1 + \frac{1}{4}J^1(L) + \frac{1}{4}J^1(M) + \frac{1}{4}J^1(R) \\
&\overset{\text{using (3) and (4)}}{=} 2 + \frac{1}{2}J^1(M) = 4 \\
&\Rightarrow J^1(L) = J^1(M) = J^1(R) = 4
\end{aligned}$$

Policy improvement:

$$\begin{aligned}
\mu^1(L) &= \underset{u\in U(L)}{\operatorname{argmin}} \left[1 + p_{LL}(a_L)J^1(L) + p_{LM}(a_L)J^1(M), 1 + p_{LL}(b_L)J^1(L) + p_{LM}(b_L)J^1(M)\right] \\
&= \underset{u\in U(L)}{\operatorname{argmin}} [4, 5] = a_L \\
\mu^1(R) &= \underset{u\in U(R)}{\operatorname{argmin}} \left[1 + p_{RR}(a_R)J^1(R) + p_{RM}(a_R)J^1(M), 1 + p_{RR}(b_R)J^1(R) + p_{RM}(b_R)J^1(M)\right] \\
&= \underset{u\in U(R)}{\operatorname{argmin}} [4, 3] = b_R \\
\mu^1(M) &= \underset{u\in U(M)}{\operatorname{argmin}} \big[1 + p_{ML}(a_M)J^1(L) + p_{MM}(a_M)J^1(M) + p_{MR}(a_M)J^1(R), \\
&\qquad 1 + p_{ML}(b_M)J^1(L) + p_{MM}(b_M)J^1(M) + p_{MR}(b_M)J^1(R), \\
&\qquad 1 + p_{ML}(c_M)J^1(L) + p_{MM}(c_M)J^1(M) + p_{MR}(c_M)J^1(R)\big] \\
&= \underset{u\in U(M)}{\operatorname{argmin}} [3, 2, 4] = b_M
\end{aligned}$$

Iteration 2:

Policy evaluation:

$$
\begin{aligned}
J^2(L) &= 1 + p_{LL}(a_L)J^2(L) + p_{LM}(a_L)J^2(M) \\
&= 1 + \frac{1}{2}J^2(L) + \frac{1}{4}J^2(M) \\
&\Rightarrow J^2(L) = 2 + \frac{1}{2}J^2(M) \qquad (5) \\
J^2(R) &= 1 + p_{RR}(b_R)J^2(R) + p_{RM}(b_R)J^2(M) \\
&\Rightarrow J^2(R) = 1 + \frac{1}{2}J^2(M) \qquad (6) \\
J^2(M) &= 1 + p_{ML}(b_M)J^2(L) + p_{MM}(b_M)J^2(M) + p_{MR}(b_M)J^2(R) \\
&\overset{\text{using (5) and (6)}}{=} 1 + \frac{1}{4}J^2(M) \\
&\Rightarrow J^2(M) = \frac{4}{3},\ J^2(L) = \frac{8}{3}, J^2(R) = \frac{5}{3}
\end{aligned}
$$

Policy improvement:

$$
\begin{aligned}
\mu^2(L) &= \underset{u\in U(L)}{\operatorname{argmin}} \left[1 + p_{LL}(a_L)J^2(L) + p_{LM}(a_L)J^2(M), 1 + p_{LL}(b_L)J^2(L) + p_{LM}(b_L)J^2(M)\right] \\
&= \underset{u\in U(L)}{\operatorname{argmin}} \left[\frac{8}{3}, \frac{9}{3}\right] = a_L \\
\mu^2(R) &= \underset{u\in U(R)}{\operatorname{argmin}} \left[1 + p_{RR}(a_R)J^2(R) + p_{RM}(a_R)J^2(M), 1 + p_{RR}(b_R)J^2(R) + p_{RM}(b_R)J^2(M)\right] \\
&= \underset{u\in U(R)}{\operatorname{argmin}} \left[\frac{13}{6}, \frac{10}{6}\right] = b_R \\
\mu^2(M) &= \underset{u\in U(M)}{\operatorname{argmin}} \left[1 + p_{ML}(a_M)J^2(L) + p_{MM}(a_M)J^2(M) + p_{MR}(a_M)J^2(R),\right. \\
&\qquad 1 + p_{ML}(b_M)J^2(L) + p_{MM}(b_M)J^2(M) + p_{MR}(b_M)J^2(R), \\
&\qquad \left.1 + p_{ML}(c_M)J^2(L) + p_{MM}(c_M)J^2(M) + p_{MR}(c_M)J^2(R)\right] \\
&= \underset{u\in U(M)}{\operatorname{argmin}} \left[\frac{25}{12}, \frac{16}{12}, \frac{29}{12}\right] = b_M
\end{aligned}
$$

Iteration 3:

Policy evaluation:

$$\Rightarrow J^3(M) = \frac{4}{3},\ J^3(L) = \frac{8}{3}, J^3(R) = \frac{5}{3}$$

Since $J^3(i) = J^2(i)$ holds for all nodes i the policy iteration algorithm has converged.

b) **Iteration 1:**

$$\begin{aligned}
J^1(L) &= \min\left[1 + p_{LL}(a_L)J^0(L) + p_{LM}(a_L)J^0(M), 1 + p_{LL}(b_L)J^0(L) + p_{LM}(b_L)J^0(M)\right] \\
&= \min[4, 5] = 4 \Rightarrow u^1(L) = a_L \\
J^1(R) &= \min\left[1 + p_{RR}(a_R)J^0(R) + p_{RM}(a_R)J^0(M), 1 + p_{RR}(b_R)J^0(R) + p_{RM}(b_R)J^0(M)\right] \\
&= \min[4, 3] = 3 \Rightarrow u^1(R) = b_R \\
J^1(M) &= \min\left[1 + p_{MR}(a_M)J^0(R) + p_{MM}(a_M)J^0(M) + p_{ML}(a_M)J^0(L),\right. \\
&\qquad\qquad 1 + p_{MR}(b_M)J^0(R) + p_{MM}(b_M)J^0(M) + p_{ML}(b_M)J^0(L), \\
&\qquad\qquad \left. 1 + p_{MR}(c_M)J^0(R) + p_{MM}(c_M)J^0(M) + p_{ML}(c_M)J^0(L)\right] \\
&= \min[3, 2, 4] = 2 \Rightarrow u^1(R) = b_M
\end{aligned}$$

Iteration 2:

$$\begin{aligned}
J^2(L) &= \min\left[1 + p_{LL}(a_L)J^1(L) + p_{LM}(a_L)J^1(M), 1 + p_{LL}(b_L)J^1(L) + p_{LM}(b_L)J^1(M)\right] \\
&= \min\left[\frac{7}{2}, \frac{8}{2}\right] = \frac{7}{2} \Rightarrow u^2(L) = a_L \\
J^2(R) &= \min\left[1 + p_{RR}(a_R)J^1(R) + p_{RM}(a_R)J^1(M), 1 + p_{RR}(b_R)J^1(R) + p_{RM}(b_R)J^1(M)\right] \\
&= \min[3, 2] = 2 \Rightarrow u^2(R) = b_R \\
J^2(M) &= \min\left[1 + p_{MR}(a_M)J^1(R) + p_{MM}(a_M)J^1(M) + p_{ML}(a_M)J^1(L),\right. \\
&\qquad\qquad 1 + p_{MR}(b_M)J^1(R) + p_{MM}(b_M)J^1(M) + p_{ML}(b_M)J^1(L), \\
&\qquad\qquad \left. 1 + p_{MR}(c_M)J^1(R) + p_{MM}(c_M)J^1(M) + p_{ML}(c_M)J^1(L)\right] \\
&= \min\left[\frac{11}{4}, \frac{6}{4}, \frac{13}{4}\right] = \frac{3}{2} \Rightarrow u^2(R) = b_M
\end{aligned}$$

c) Value iteration requires the definition of a stopping criterion, e.g. $J^k(i) - J^{k-1}(i) < \epsilon$ for all nodes i. If this stopping criterion is fulfilled, the algorithm is terminated. However, it is not guaranteed that the policy has converged.