Solutions of Problem set 1
Optimal control and reinforcement learning, 4SC000, TU/e

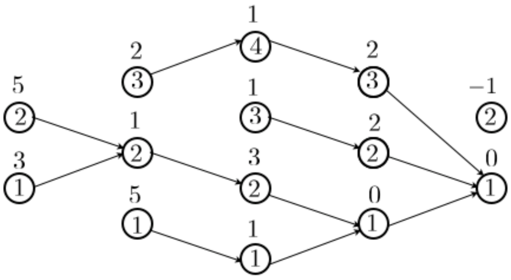
Duarte Antunes

Q2, 2022-2023

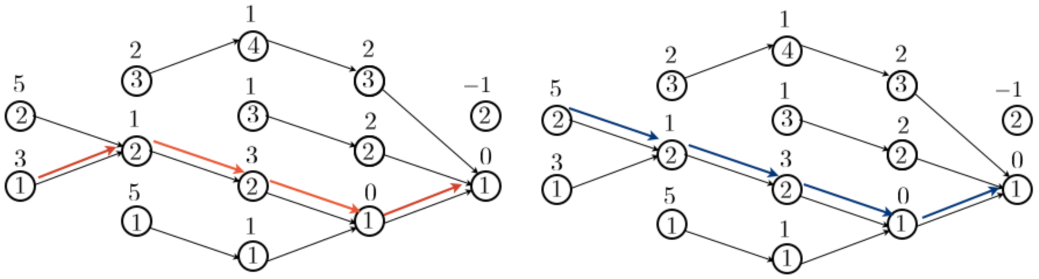
1. Modeling and the dynamic programming algorithm

Problem 1.1 Let us start with the transition diagram (a)

(i) Using the dynamic programming algorithm we obtain a unique policy indicated in the figure along with the costs-to-go



The optimal paths can be obtained by following the decisions of the optimal policy and are indicated in the next figure (on the left for $x_0 = 1$ on the right for $x_0 = 2$). Specifying an optimal path graphically is the easiest way.



Alternatively, we can specify the optimal path by providing the associated decisions and states

$$\{(x_0, u_0), (x_1, u_1), (x_2, u_2), (x_3, u_3), (x_4)\}.$$

For this, we need to label the possible decisions for each state (the states are already labeled). Following the convention used in class we label these possible decisions as in Figure 2. Then an optimal path which starts at $x_0 = 1$ is

$$\{(1, 2), (2, 1), (2, 1), (1, 1), (1)\}.$$

and an optimal path which starts at $x_0 = 2$ is

$$\{(2,1),(2,1),(2,1),(1,1),(1)\}.$$

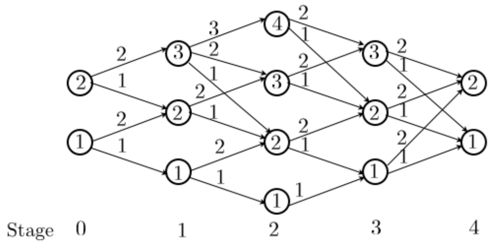


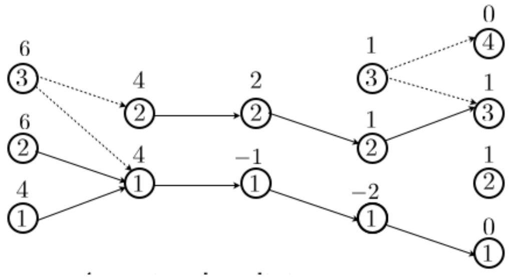
Figure 1: Labeling decisions

(ii) This was solved already in (i).

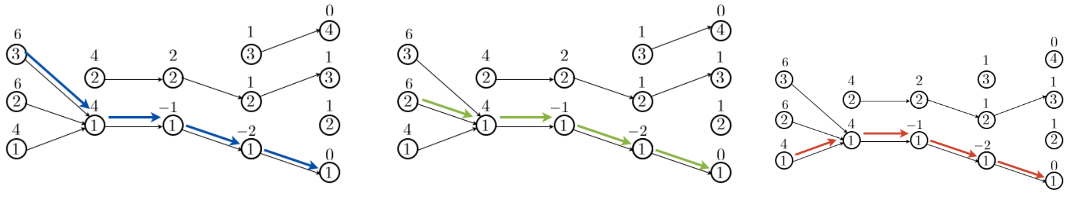
(iii) For the transition diagram (a) the optimal policy and the optimal paths are unique.

Consider now the transition diagram (b).

(i) Applying the dynamic programming algorithm we obtain the costs-to-go and possible optimal decisions indicated in the figure below Note that for state 4 at stage 3 there are two possible



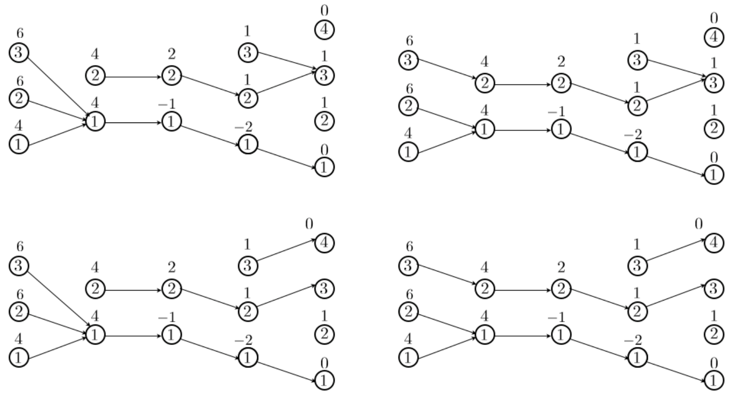
optimal decisions as well as for state 3 at stage 0. This results from the dynamic programming algorithm for which more than one decision results in the same cost-to-go for these states. One optimal path for each possible initial state is indicated in the figure below. Note that for state



$x_0 = 3$ we could have chosen another optimal path.

(ii) This was solved already in (i).

(iii) For the transition diagram (b) the optimal policy and the optimal paths are not unique. The number of optimal paths for initial state $x_0 = 3$ is 2 and the number of optimal paths for states $x_0 = 1$ and $x_0 = 2$ is one. The number of optimal polices is 4 and are indicated in the figure below. The costs-to-go are always the same for different optimal policies.



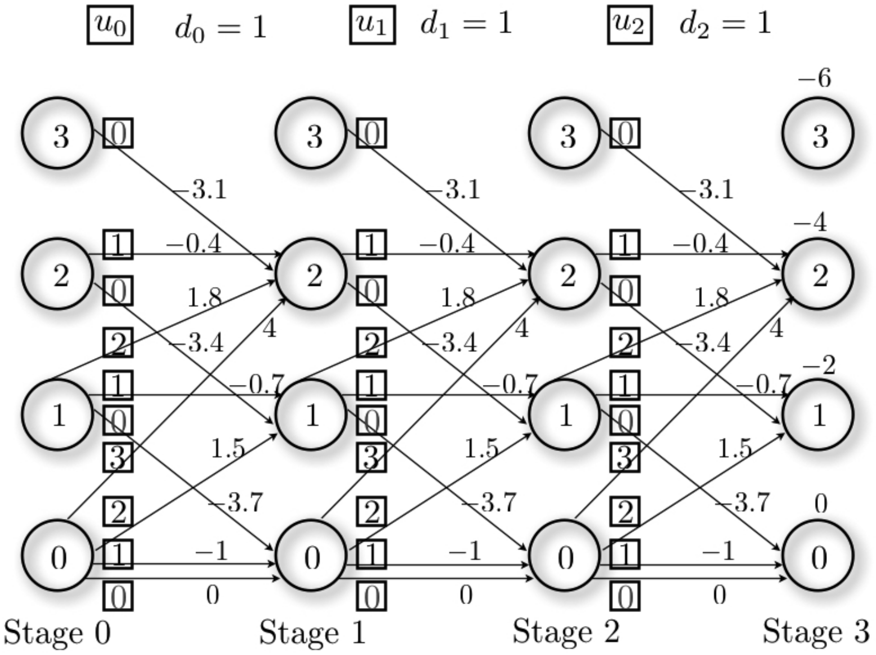


Figure 2: Transition diagram

Problem 1.2 The transition diagram is shown in Fig. 2 and the result of applying the dynamic programming algorithm is shown in Fig. 3. For example at stage 2, state 1, the optimal decision is $u_2 = 0$ because the following minimization

$$
\begin{aligned}
J_2(1) &= \min_{u_2} g_2(1, u_k) + J_3(\max\{1 + u_k - 1, 0\}) \\
&= \min\{g_2(1,0) + J_3(0), g_2(1,1) + J_3(1), g_2(1,2) + J_3(2)\} \\
&= \min\{-3.7 + 0, -0.7 - 2, 1.8 - 4\}) = -3.7
\end{aligned}
$$

corresponds to $u_2 = 0$, where $J_3(x_3) = g_3(x_3)$ and

$$
g_k(x_k, u_k) = \big(c_1(x_k) + c_2(u_k) - p\min\{x_k + u_k, d_k\}\big).
$$

This also gives the value of the cost-to-go $J_2(1)$. After applying the dynamic programming algorithm, it is clear that there are $N = 2$ optimal policies described as follows

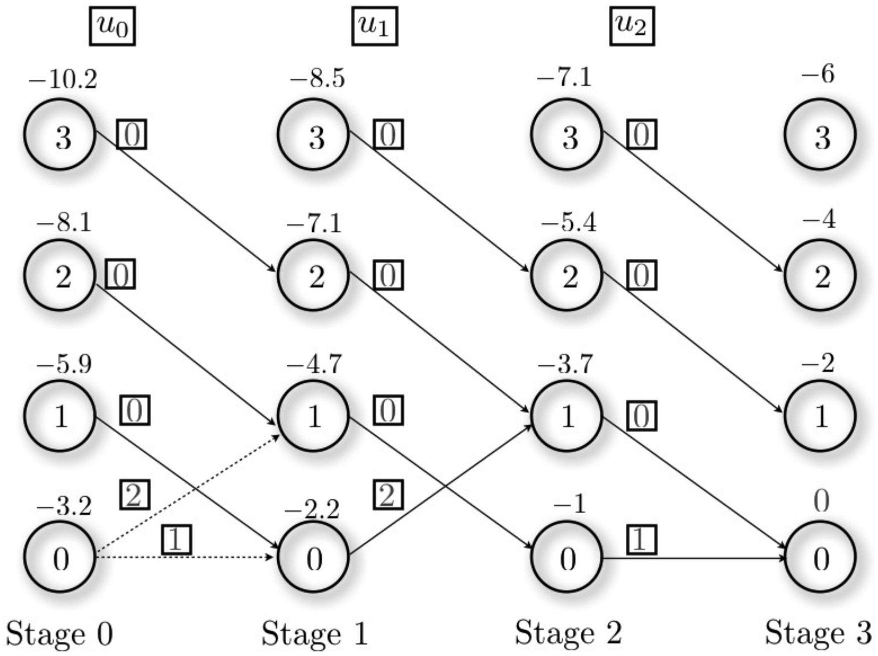


Figure 3: Optimal policy and costs-to-go

	$u_0 = \mu_0(x_0)$	$u_1 = \mu_1(x_1)$	$u_2 = \mu_2(x_2)$
3	0	0	0
2	0	0	0
1	0	0	0
0	2	2	1
x_k	$k = 0$	$k = 1$	$k = 2$

	$u_0 = \mu_0(x_0)$	$u_1 = \mu_1(x_1)$	$u_2 = \mu_2(x_2)$
3	0	0	0
2	0	0	0
1	0	0	0
0	1	2	1
x_k	$k = 0$	$k = 1$	$k = 2$

and the optimal costs-to-go $J_k(x_k)$ are

	$J_0(x_0)$	$J_1(x_1)$	$J_2(x_2)$
3	-10.2	-8.5	-7.1
2	-8.1	-7.1	-5.4
1	-5.9	-4.7	-3.7
0	-3.2	-2.2	-1
x_k	$k = 0$	$k = 1$	$k = 2$

Problem 1.3

(i) The optimal policies are summarized in Figure 4. There are two optimal policies since there are two possible optimal decisions at stage 1 state -1. The optimal supplies x_0 are obtained by following the arrows for each initial state (each arrow is associated with an action). The optimal total costs are the costs-to-go at the initial states. Thus,

- For $x_0 = 0$, the optimal supplies are $u_0 = 0$, $u_1 = 0$, $u_2 = 3$ and result in a cost 3.35. Moreover, $u_0 = 0$, $u_1 = 3$, $u_2 = 0$ are also optimal suplies and result in the same cost.
- For $x_0 = 1$, the optimal supplies are $u_0 = 0$, $u_1 = 0$, $u_2 = 2$ and result in a cost 2.35.
- For $x_0 = 2$, the optimal supplies are $u_0 = 0$, $u_1 = 0$, $u_2 = 1$ and result in a cost 1.5.

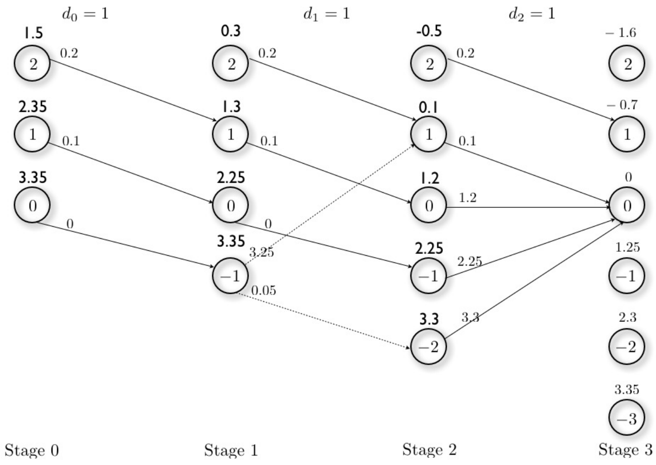


Figure 4: Optimal policy $c_{\text{tr}} = 0.2$

Open-loop supplies $\bar{u}_0 = 0$, $\bar{u}_1 = 0$, $\bar{u}_2 = 3$.

(ii) The new transition diagram for c_{tr} is depicted in Figure 5, whereas the optimal policy is depicted in Figure 6. The policy is different and since the transportation cost is smaller there is a tendency in the optimal policy to set the inventory to zero (reducing costs incurred with either positive or negative stock).

(iii) We have obtained two optimal paths in (i) with supplies $\bar{u}_0 = 0$, $\bar{u}_1 = 0$, $\bar{u}_2 = 3$ and $\bar{u}_0 = 0$, $\bar{u}_1 = 0$, $\bar{u}_2 = 3$. Let us consider the two cases separately.

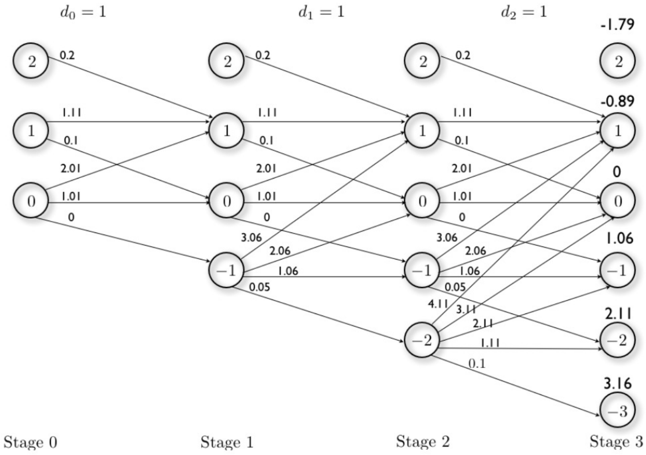


Figure 5: Transition diagram for $c_{\text{tr}} = 0.01$

Open-loop If $d_1 = 1$, we obtain exactly the same cost as in (i) that is 3.35. If $d_1 = 0$ we notice that if we apply $\bar{u}_0 = 0$, $\bar{u}_1 = 0$, $\bar{u}_2 = 3$ the state will be $x_0 = 0$, $x_1 = -1$, $x_2 = -1$, $x_3 = 1$ since

$$x_{k+1} = x_k + u_k - d_k$$

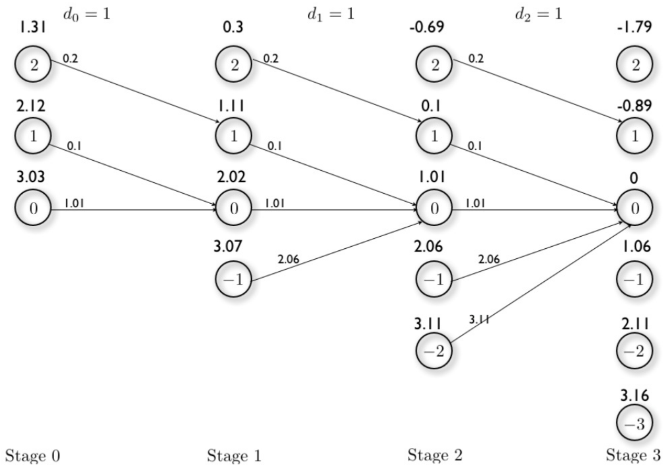


Figure 6: Optimal policy for $c_{\text{tr}} = 0.01$

and $d_0 = 1$,$d_1 = 1$, $d_2 = 0$. Then the cost is

$$\underbrace{c_1(0) + c_2(0)}_{=0} + \underbrace{c_1(-1) + c_2(0)}_{=0.05} + \underbrace{c_1(-1) + c_2(3)}_{=3.25} + \underbrace{g_3(1)}_{=-0.7} = 2.6$$

Since Prob$[d_1 = 1] = 0.6$, Prob$[d_1 = 0] = 0.4$, the expected cost is then

$$0.6 \times 3.35 + 0.4 \times 2.6 = 3.05$$

Closed-loop If $d_1 = 1$, we obtain exactly the same cost as in (i) that is 3.35. If $d_1 = 0$, if we readjust the decisions to cope with uncertainty based on the optimal policy (computed neglecting disturbances in (i)) we see that the decisions at stages 0 and 1 are the same (just follows the policy in (i)) for $x_0 = 0$ the decision is $u_0 = 0$ and for $x_1 = -1$ the decision we are assuming in this case is $u_1 = 0$). However, after the disturbance occurs and the state at stage 2 is $x_2 = -1$ if we use the decision specified by the optimal policy we obtain $u_2 = 2$ instead of the open loop one ($u_2 = 3$). Then the final state will be $x_3 = 0$. The cost is given by

$$\underbrace{c_1(0) + c_2(0)}_{=0} + \underbrace{c_1(-1) + c_2(0)}_{=0.05} + \underbrace{c_1(-1) + c_2(2)}_{=2.25} + \underbrace{g_3(0)}_{=0} = 2.3.$$

Then the expected cost is

$$0.6 \times 3.35 + 0.4 \times 2.3 = 2.93.$$

The expected cost of the closed-loop is smaller. This follows from the fact that at the final stage the decision is adapted to minimize the cost-to-go.

Open-loop supplies $\bar{u}_0 = 0$, $\bar{u}_1 = 3$, $\bar{u}_2 = 0$.

Open-loop If $d_1 = 1$, we obtain exactly the same cost as in (i) that is 3.35. If $d_1 = 0$ we notice that if we apply $\bar{u}_0 = 0$, $\bar{u}_1 = 3$, $\bar{u}_2 = 0$ the state will be $x_0 = 0$, $x_1 = -1$, $x_2 = 2$, $x_3 = 1$. Then the cost is Then the cost is

$$\underbrace{c_1(0) + c_2(0)}_{=0} + \underbrace{c_1(-1) + c_2(3)}_{=3.25} + \underbrace{c_1(2) + c_2(0)}_{=0.2} + \underbrace{g_3(1)}_{=-0.7} = 2.75$$

Since Prob$[d_1 = 1] = 0.6$, Prob$[d_1 = 0] = 0.4$, the expected cost is then

$$0.6 \times 3.35 + 0.4 \times 2.75 = 3.11$$

Closed-loop If $d_1 = 1$, we obtain exactly the same cost as in (i) that is 3.35. If $d_1 = 0$, if we readjust the decisions to cope with uncertainty based on the optimal policy (computed neglecting disturbances in (i)) we see that the decisions at stages 0 and 1 are the same (just follows the policy in (i)) for $x_0 = 0$ the decision is $u_0 = 0$ and for $x_1 = -1$ the decision we are assuming in this case is $u_1 = 0$). Moreover, after the disturbance occurs and the state at stage 2 is $x_2 = 2$ if we use the decision specified by the optimal policy we obtain $u_2 = 0$ exactly as for the open loop policy. Since the decisions are the same as in the open loop we obtain the same cost 2.75. The expected value is then also the same and given by

$$0.6 \times 3.35 + 0.4 \times 2.75 = 3.11.$$

The expected cost of the closed-loop is the same since the decisions are the same (after the disturbance at stage 1, the decision did not change at stage 3).

Problem 1.4

a. This is a continuous-time optimal control problem with dynamic model

$$\dot{\underline{x}} = A\underline{x} + Bu$$

$$\underline{x} = \begin{bmatrix} x \\ \dot{x} \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{k_f}{m} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix},$$

where m is the mass and $\frac{k_f}{m}$ is the friction coefficient. The initial condition is $x(0) = 0$, $v(0) = \bar{v}_0$, for some constant $\bar{v}_0$, and there is a terminal constraint $x(T) = 1$. The cost function (to be minimized) is

$$\int_0^T g(x(t), u(t))dt = T$$

where $g(x(t), u(t)) = 1$.

b. This is a stage decision problem. Let us assume that

$$u(t) = u_k, \quad t \in [t_k, t_{k+1})$$

and consider $\underline{x}_k := x(k\tau)$, $k \in \mathbb{N}_{\geq 0}$ where τ is the sampling period. Then the dynamic model is

$$\underline{x}_{k+1} = A_d \underline{x}_k + B_d u_k$$

where $A_d = e^{A\tau}$, $B_d = \int_0^\tau e^{As} ds B$. The initial condition is $x_0 = 0$ and $v_0 = \bar{v}_0$. The terminal constraint is $x_n = 1$, where $x_h = x(h\tau)$. The cost function is again the final time, in this case discrete-time h.

c. This is a discrete optimization problem. A small trick allows to formulate it directly in the framework of a transition diagram: for each node count the maximum number of hops required to arrive at that node from the Start node. Set the stage of that node to be this maximum number. If the outgoing arrow of a given node at stage k leads to a node which is at a stage greater than $k+1$, say ℓ, add fictitious states at stages $k+2, \dots, \ell - 1$ whose outgoing arrows have cost zero. The cost of the outgoing arrow for that node at stage k, connecting now to a fictitious state at stage $k+1$ should remain unchanged. This is illustrated in Figure 7 for the given diagram. One can now obtain the critical path by running the DP algorithm using the max iteration instead of the min operation (or multiplying all the gains by -1 and running the usual DP algorithm with the min operation).

d. This is a continuous-time optimal control problem. The dynamic model is

$$\dot{\underline{x}} = A\underline{x} + Bu$$

$$\underline{x} = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \frac{1}{I} \end{bmatrix},$$

where I is the moment of inertia and $u = T$. There is a terminal constraint $\theta(t_0 + h) = \theta_{\text{des}}$ and the cost function is

$$\int_{t_0}^{t_0+h} c(u(t))dt.$$

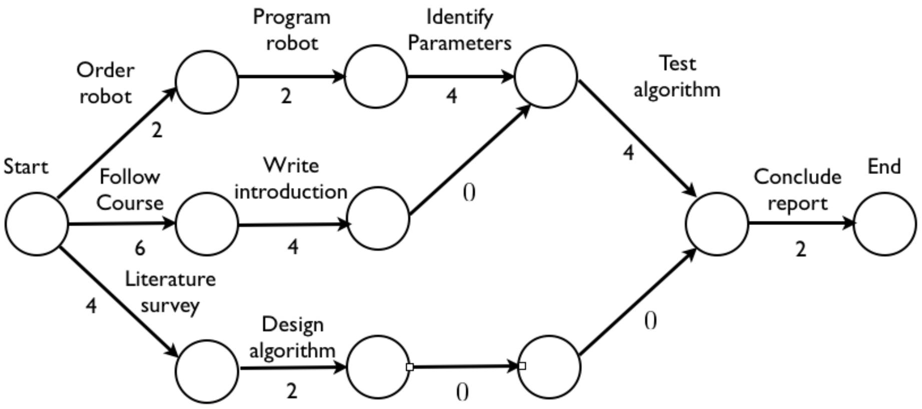


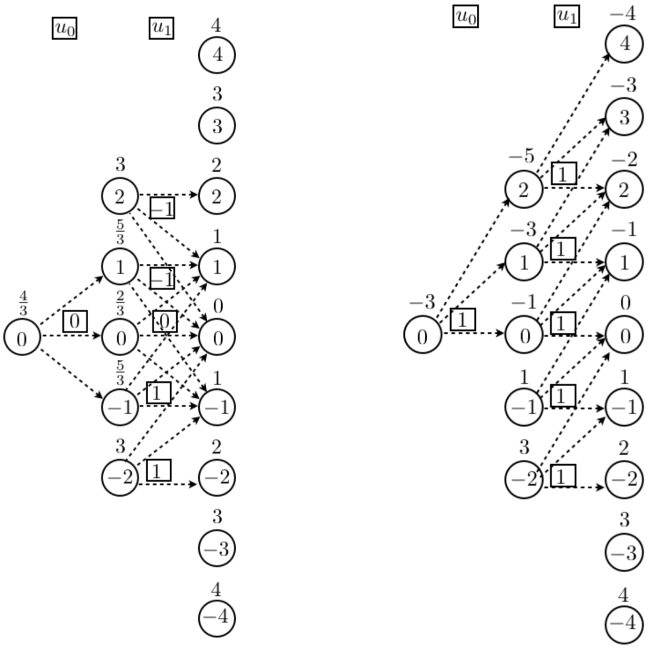
Figure 7: Transition diagram for a graph activity network

2. Stochastic dynamic programming

Problem 2.1

(i) For both problems $X_1 = \{-2, -1, 0, 1, 2\}$, $X_2 = \{-4, -3, -2, -1, 0, 1, 2, 3, 4\}$.

(ii) The optimal policies for Problems 1 and 2 are depicted in the figure and can be summarized as follows: for Problem 1 for stage 0 and 1, pick $u_k = 0$ if $x_k = 0$, pick $u_k = -1$ if $x_k > 0$, pick $u_k = 1$ if $x_k < 0$; for Problem 2 pick $u_k = 1$ for every stage and every state. The optimal policies are



(a) Prob. 1

(b) Prob. 2

Figure 8: Optimal policies

obtained by running the stochastic dynamic programming algorithm. For example for Problem 1, at stage 1, state $x_1 = -2$ there are three options $u_1 = -1$, $u_1 = 0$, $u_1 = 1$ with expected costs

$$\frac{1}{3}(|x_1| + |x_1 + u_1 - 1|) + \frac{1}{3}(|x_1| + |x_1 + u_1 + 0|) + \frac{1}{3}(|x_1| + |x_1 + u_1 + 1|).$$

This is minimized for $u_1 = 1$ and yields an expected cost of 3, as depicted in the figure.

(iii) The (overall) expected costs are the costs-to-go at stage 0, state 0, i.e., for Problem 1 the expected cost is $\frac{4}{3}$ and for Problem 2 the expected cost is -3.

Problem 2.2 (i) To apply the stochastic dynamic programming algorithm, we start at the last decision stage (state 2). Since there are no disturbances this is identical to the deterministic DP algorithm, i.e., we obtain the same decisions and costs-to-go as in stage 2 of Figure 4. We call this cost-to-go $J_2(x_2)$, $x_2 \in \{-2, -1, 0, 1, 2\}$, which is given by $J_2(-2) = 3.3$, $J_2(-1) = 2.25$, $J_2(0) = 1.2$ $J_2(1) = 0.1$, $J_2(2) = -0.5$. At stage 1, we must compute for each state $x_1 \in \{-1, 0, 1, 2\}$ what is the decision which leads to the smallest expected cost. This is summarized in the next tables. From the values in the table we conclude that for every state $x_1 \in \{-1, 0, 1, 2\}$, $u_1 = 0$ is the value which leads to the minimum cost. The associated expected costs-to-go are denoted by $J_1(x_1)$ and are depicted in Figure (9) (stage 1).

At stage 0 applying the stochastic dynamic programming algorithm is equivalent to applying the deterministic DP algorithm with the cost-to-go of stage 1 (which now are expected costs). This results in Figure 10 where the costs-to-go at stage 0 represent expected total cost.

The optimal possible decisions for state -1 at stage 1 changes. Before (deterministic DP algorithm) these were $u_1 = 3$ and $u_1 = 0$ and now only $u_1 = 0$ (in general deterministic and stochastic DP provide different optimal policies).

(ii) The expected cost for the initial condition $x_0 = 0$ coincides with the cost-to-go at stage 0 and is given by 2.93. In this particular case it coincides with the cost obtained with the deterministic DP algorithm. The value obtained with the stochastic DP algorithm must always be smaller or equal than the one obtained with the deterministic DP algorithm, since the former minimizes exactly the expected cost, whereas the latter minimizes the cost assuming no disturbances.

$x_1 = -1$

u_1	$\mathbb{E}_{d_1}[c_1(-1) + c_2(u_1) + J_2(-1 + u_1 - d_1)]$
0	$(0.05 + 3.3) \times 0.6 + (0.05 + 2.25) \times 0.4 = 2.93$
1	$(1.25 + 2.25) \times 0.6 + (1.25 + 1.2) \times 0.4 = 3.08$
2	$(2.25 + 1.2) \times 0.6 + (2.25 + 0.1) \times 3.01$
3	$(3.25 + 0.1) \times 0.6 + (3.25 - 0.5) \times 0.4 = 3.11$

$x_1 = 0$

u_1	$\mathbb{E}_{d_1}[c_1(0) + c_2(u_1) + J_2(0 + u_1 - d_1)]$
0	$(0 + 2.25) \times 0.6 + (0 + 1.2) \times 0.4 = 1.83$
1	$(1.2 + 1.2) \times 0.6 + (1.2 + 0.1) \times 0.4 = 1.96$
2	$(2.2 + 0.1) \times 0.6 + (2.2 - 0.5) \times 0.4 = 2.06$

$x_1 = 1$

u_1	$\mathbb{E}_{d_1}[c_1(1) + c_2(u_1) + J_2(1 + u_1 - d_1)]$
0	$(0.1 + 1.2) \times 0.6 + (0.1 + 0.1) \times 0.4 = 0.86$
1	$(1.3 + 0.1) \times 0.6 + (1.3 - 0.5) \times 0.4 = 1.16$

$x_1 = 2$

u_1	$\mathbb{E}_{d_1}[c_1(1) + c_2(u_1) + J_2(1 + u_1 - d_1)]$
0	$(0.2 + 0.1) \times 0.6 + (0.2 - 0.5) \times 0.4 = 0.06$

Problem 2.3

(i) The transition diagram resulting from the values given in the exercise is shown in Figure 11. Applying the dynamic programming algorithm we obtain the optimal policies shown in Figure 12 (there are two optimal policies because there are two optimal actions for state 0 at stage 0). If $x_0 = 2$, following the arrows, we obtain that the optimal supplies are $u_0 = 0$, $u_1 = 0$, $u_2 = 1$.

(ii).(a) For the supplies $u_0 = 0$, $u_1 = 0$, $u_2 = 1$, the cost as a function of d_1 is

$$\underbrace{c_1(2)}_{=1} + \underbrace{c_2(0)}_{=0} - \underbrace{p\min\{2,1\}}_{=5} + \underbrace{c_1(1)}_{=0.5} + \underbrace{c_2(0)}_{=0} - p\min\{1, d_1\} + c_1(\max\{1 - d_1, 0\}) + \underbrace{c_2(1)}_{=2.8} - \underbrace{p}_{=5} + g_3(x_3)$$

For:

$d_1 = 0$ $x_2 = \max\{1 - d_1, 0\} = 1$, $x_3 = 1$ and the cost is -7

$d_1 = 1$ $x_2 = \max\{1 - d_1, 0\} = 0$, $x_3 = 0$ and the cost is -10.7

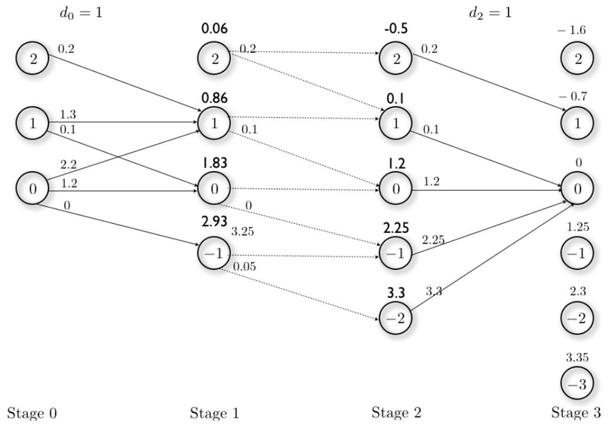


Figure 9: Step 2 of the dynamic programming algorithm

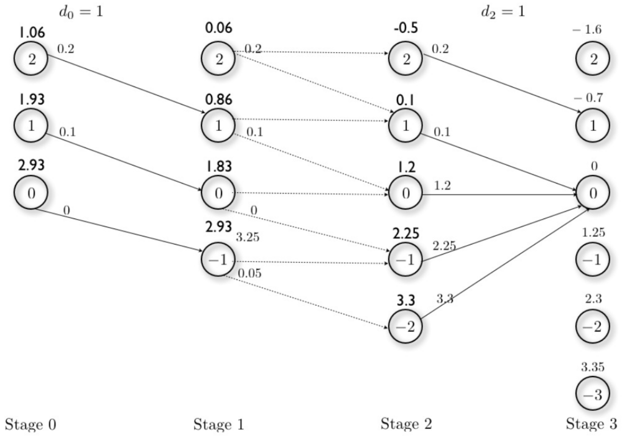


Figure 10: Optimal policy and costs-to-go resulting from the stochastic DP algorithm

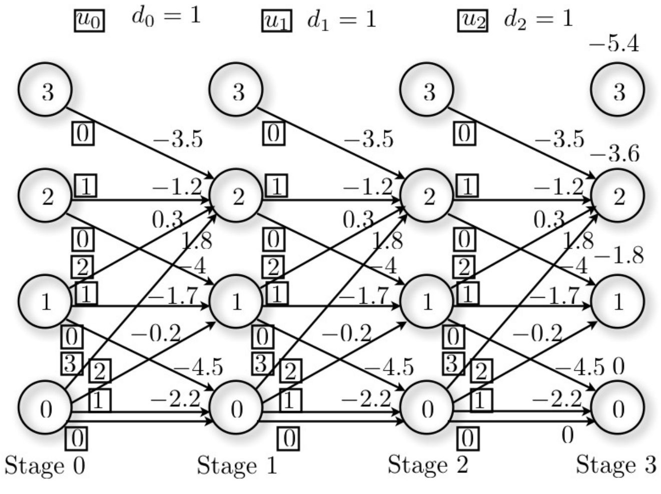


Figure 11: Transition diagram

$d_1 = 2$ $x_2 = \max\{1 - d_1, 0\} = 0$, $x_3 = 0$ and the cost is -10.7

Thus, the expected cost is $-7 \times 0.4 + -10.7 \times 0.4 - 10.7 \times 0.2 = -9.22$.

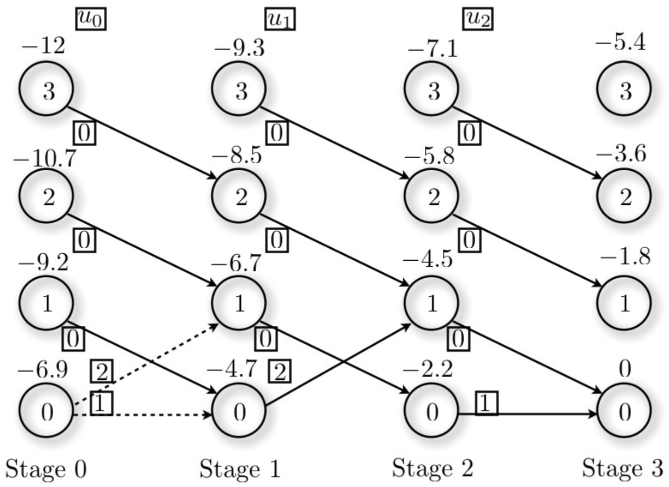


Figure 12: Optimal policies

(ii).(b) Taking into account the optimal policy computed in (i) we see that if at stage 2 the state is $x_2 = 0$ the decision is the same $u_2 = 1$ as the open-loop decision. Thus

$d_1 = 1$ $x_2 = \max\{1 - d_1, 0\} = 1$, $x_3 = 0$ and the cost is -10.7

$d_1 = 2$ $x_2 = \max\{1 - d_1, 0\} = 1$, $x_3 = 0$ and the cost is -10.7

However, if the state is $x_2 = 1$ the decision for the policy computed in (i) is $u_2 = 0$ and $x_3 = 0$ this means that the cost is

$$\underbrace{c_1(2)}_{=1}+\underbrace{c_2(0)}_{=0}-\underbrace{p\min\{2,1\}}_{=5}+\underbrace{c_1(1)}_{=0.5}+\underbrace{c_2(0)}_{=0}-\underbrace{p\min\{1,d_1\}}_{=0}+\underbrace{c_1(\max\{1-d_1,0\})}_{=0.5}+\underbrace{c_2(0)}_{=0}-\underbrace{p}_{=5}+\underbrace{g_3(0)}_{=0}=-8$$

Thus, the expected cost is

$$-8 \times 0.4 + -10.7 \times 0.2 + -10.7 \times 0.4 = -9.62$$

(ii).(c) To apply the stochastic dynamic programming algorithm we start by noticing that at stage 2 there are no disturbances and therefore the costs-to-go coincide with the ones computed in (i).

Stage 1

We have to compute for every state $x_1 \in \{0, 1, 2, 3\}$ the optimal decision, i.e., the one that minimizes the expected cost corresponding to that decision. We illustrate this for stage $x_1 = 1$ in Figure 13. The costs-to-go obtained with each decision are

$x_1 = u_1 = 0$ $0.4(0.5 - 4.5) + 0.2(-4.5 - 2.2) + 0.4(-4.5 - 2.2) = -5.62$

$u_1 = 1$ $0.4(3.3 - 5.8) + 0.2(-4.5 - 1.7) + 0.4(-6.7 - 2.2) = -5.8$

$u_1 = 2$ $0.4(5.5 - 7.3) + 0.2(-5.8 + 0.3) + 0.4(-4.7 - 4.5) = -5.5$

and therefore the optimal decision is $u_1 = 1$ and leads to a cost-to-go of -5.8. A similar reasoning for the remaining states leads to

$x_1 = 0$: Expected cost -4.3 (optimal decision is $u_1 = 2$)

$u_1 = 0$ 0

$u_1 = 1$ $0.4(-2.2 - 2.2) + 0.2(-2.2 - 2.2) + 0.4(-4.5 + 2.8) = -3.32$

$u_1 = 2$ $0.4(-5.2 - 2.2) + 0.2(-0.2 - 4.5) + 0.4(4.8 - 5.8) = -4.3$

$u_1 = 3$ $0.4(6.8 - 7.1) + 0.2(1.8 - 5.8) + 0.4(-3.2 - 4.5) = -4$

$x_1 = 2$: Expected cost -8.1 (optimal decision is $u_1 = 0$)

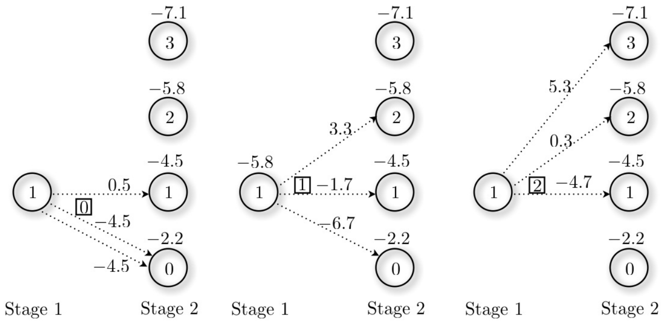


Figure 13: Computing the optimal decision for $x_1 = 1$ (stochastic DP)

$$u_1 = 1 \;\; 0.4(3.8 - 7.1) + 0.2(-1.2 - 5.8) + 0.4(-6.2 - 4.5) = -7$$
$$u_1 = 0 \;\; 0.4(1 - 5.8) + 0.2(-4.5 - 4) + 0.4(-9 - 2.2) = -8.1$$

$x_1 = 3$: Expected cost -7.74 (optimal decision is $u_1 = 0$)

$$u_1 = 0 \;\; 0.4(-3.5 - 3.6) + 0.2(1.5 - 5.4) + 0.4(-8.5 - 1.8) = -7.74$$

Stage 0

At stage 0 there are no disturbances and therefore it suffices to apply one iteration of the deterministic DP algorithm with the **expected** costs-to-go computed at stage 1. For example for $x_0 = 1$, we take the following minimization with respect to the decisions $u_0 = 0$, $u_0 = 1$ $u_0 = 2$, respectively

$$\min(\begin{bmatrix}-4.5 & -1.7 & 0.3\end{bmatrix} + \begin{bmatrix}-4.3 & -5.8 - 8.1\end{bmatrix})$$

obtaining that the optimal decision is $u_0 = 0$ corresponding to an expected cost of -8.8. The optimal policy and the costs-to-go are depicted in Figure 14.

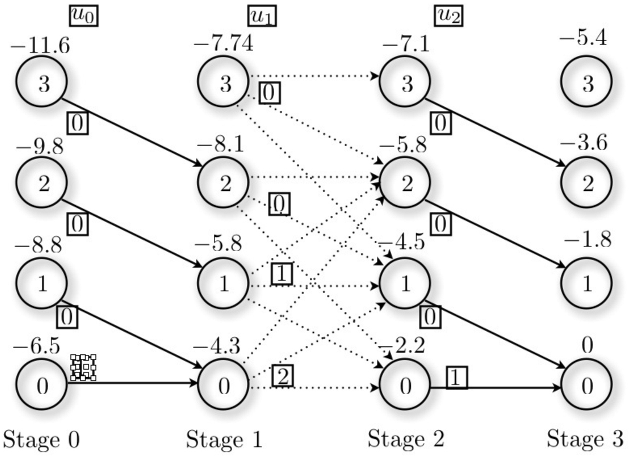


Figure 14: Optimal policy (stochastic DP)

Problem 2.4

(i) Let the state $x_k \in \{8, 9, 10, 11, 12\}$ be the offer of car dealer 1, 2, 3 at stages $k \in \{0, 1, 2\}$, respectively. Let us define an auxiliary state $x_k = t$ for stages $k \in \{1, 2\}$ to denote the fact that the lady might have accepted an offer already before stage $k \in \{1, 2\}$. There are two options at stages $k \in \{0, 1\}$, to buy the car (stop, $u_k = 1$) or not buy the car (do not stop, $u_k = 0$), and the terminal cost at stage $k = 2$ is $g_2(x_k) = x_k$. For state $x_1 = t$ there is only one option which is to go to state $x_2 = t$ (with zero cost) and the terminal cost at for state $x_2 = t$ is $g_2(t) = 0$. Accepting the offer has a cost of x_k and leads to state $x_{k+1} = t$, rejecting the offer has no (stage) cost and leads to a state $x_{k+1} \in \{8, 9, 10, 11, 12\}$ with probabilities $\{\frac{1}{3}, \frac{1}{12}, \frac{1}{6}, \frac{1}{12}, \frac{1}{3}\}$, respectively. Applying the stochastic dynamic programming algorithm we obtain

Stage 2

$$J_2(x_k) = x_k$$

Stage 1

$$\begin{aligned}J_1(x_k) &= \min\{x_k + J_2(t), \frac{1}{3}J_2(8) + \frac{1}{12}J_2(9) + \frac{1}{6}J_2(10) + \frac{1}{12}J_2(11) + \frac{1}{3}J_2(12)\}\\ &= \min\{x_k, \frac{1}{3}8 + \frac{1}{12}9 + \frac{1}{6}10 + \frac{1}{12}11 + \frac{1}{3}12\}\\ &= \min\{x_k, 10\}\end{aligned}$$

which leads to

$$J_1(8) = 8, J_1(9) = 9, J_1(10) = 10, J_1(11) = 10, J_1(12) = 10$$

and

$$u_1(8) = 1, u_1(9) = 1, u_1(10) \in \{0, 1\}, u_1(11) = 0, u_1(12) = 0$$

This means that after the offer of dealer 2, the lady should accept the offer only if it is less than or equal to 10, and if the offer is 10 rejecting has the same expected cost.

Stage 0

$$\begin{aligned}J_0(x_k) &= \min\{x_0 + J_1(t), \frac{1}{3}J_(8) + \frac{1}{12}J_1(9) + \frac{1}{6}J_1(10) + \frac{1}{12}J_1(11) + \frac{1}{3}J_1(12)\}\\ &= \min\{x_k, \frac{1}{3}8 + \frac{1}{12}9 + \frac{1}{6}10 + \frac{1}{12}10 + \frac{1}{3}10\}\\ &= \min\{x_k, 9.25\}\end{aligned}$$

which leads to

$$J_0(8) = 8, J_0(9) = 9, J_0(10) = 9.25, J_0(11) = 9.25, J_0(12) = 9.25$$

and

$$u_0(8) = 1, u_1(9) = 1, u_1(10) = 0, u_1(11) = 0, u_1(12) = 0$$

This means that after the offer of dealer 1, the lady should accept the offer only if it is less than or equal to 9. The policy is depicted in Figure 15

(ii) Given $J_0(x_k)$ obtained in (i) we have that the expected cost of the price of the car is

$$\frac{1}{3}8 + \frac{1}{12}9 + \frac{1}{6}9.25 + \frac{1}{12}9.25 + \frac{1}{3}9.25 = 8.8125$$

and accepting whichever offer of dealer 1 has an expected cost $\frac{1}{3}8 + \frac{1}{12}9 + \frac{1}{6}10 + \frac{1}{12}11 + \frac{1}{3}12 = 10$.

Problem 2.5

(i) Let h be the total number of parking places. At each stage $\ell \in \{0, \ldots, h-1\}$ corresponding to parking place $\ell+1$, which is $k = h-\ell$ places from the destination we consider a state $x_\ell \in \{F, O, S\}$ which takes the value F if the parking spot ℓ is free and takes the value O is the parking place ℓ is occupied. If the car has already been parked at a stage $r < \ell$ there are no more decisions to be taken and the state takes the value $x_\ell = S$. We can consider that at state 0, $x_\ell \in \{F, O\}$. There is a terminal stage $x_h = E$ denoting that the game has ended. At each state $x_\ell = F$ the driver can choose to park $u_\ell = P$ $(x_{k+1} = S)$ or not to park $u_\ell = DP$ $(x_{k+1} \in \{F, O\})$. Parking incurs in a cost $k = h - \ell$, whereas not parking has zero stage cost. If the decision is to not park at the final stage the cost is C. The transition diagram is depicted in Figure 16.

The stochastic dynamic programming algorithm for this problem can be written as follows

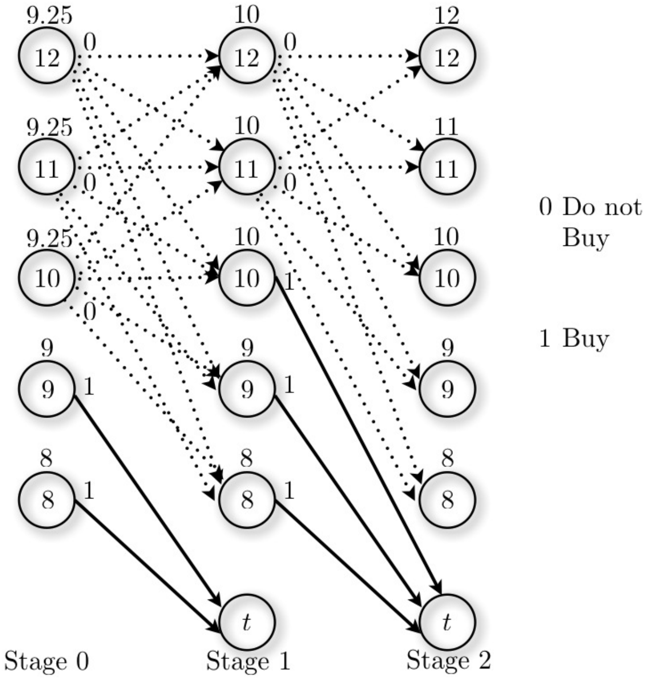


Figure 15: Optimal policy

Stage h: $J_h(E) = 0$.

Stage $h-1$: $J_{h-1}(O) = C$, $J_{h-1}(F) = \min\{C, 1\}$

Stage $h-2$:

$$\begin{aligned} J_{h-2}(O) &= pJ_{h-1}(F) + qJ_{h-1}(O) = p\min\{C,1\} + qC, \\ J_{h-2}(F) &= \min\{pJ_{h-1}(F) + qJ_{h-1}(O), 2\} = \min\{p\min\{C,1\} + qC, 2\} \end{aligned}$$

...

Stage $h-(k+1)$:

$$\begin{aligned} J_{h-(k+1)}(O) &= pJ_{h-k}(F) + qJ_{h-k}(O) \\ &= p\min\{pJ_{h-(k-1)}(F) + qJ_{h-(k-1)}(O), k\} + q(pJ_{h-(k-1)}(F) + qJ_{h-(k-1)}(O)), \\ J_{h-(k+1)}(F) &= \min\{pJ_{h-k}(F) + qJ_{h-k}(O), k+1\} \end{aligned}$$

Let $F_k := pJ_{h-k}(F) + qJ_{h-k}(O)$ for $k = \{1, 2, \ldots, h-1\}$ be the minimal expected cost if the driver is k parking places from his destination. Then from the equations for the stochastic dynamic programming algorithm we have

$$F_k = p\min\{F_{k-1}, k\} + qF_{k-1}, \quad k = \{1, 2, \ldots, h-1\}$$

for $F_0 = C$

(ii) According to the optimal policy, when $k = 1$ one picks not to park if $C < 1$. If this is the case $F_1 = qC + pC = C = F_0$ and again at $k = 2$ one picks not to park and also at $k = 3$ because $F_2 = F_1 = F_0$ and so on. In fact, if for some k one picks not to park because $F_{k-1} < k$ then $F_k = pF_{k-1} + qF_{k-1} = F_{k-1}$ and for $\ell \geq k$ one has $F_\ell = F_k = F_{k-1}$ which means that for $\ell \geq k$ one does not park. It is also clear that if $F_{k-1} > k$, i.e., the decision is to park (if a place is available) at stage $k = h - \ell$, then we must have $F_{k-2} > k-1$, i.e., the decision is to park at stage $k-1$ (otherwise from the previous reasoning, if one could not park at $k-1$ one could not park at k). By induction we conclude that for $\ell < k$ one always parks. It suffices then to find k such that

$$F_{k-1} < k \tag{1}$$

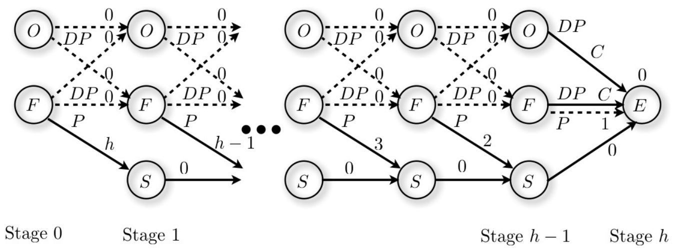


Figure 16: Transition diagram for the parking problem

knowing that for $\ell < k$ one always parks (if a place is available) and therefore $\min\{F_{\ell-1}, \ell\} = \ell$. This means that for $\ell < k$, we have

$$F_\ell = p\ell + qF_{\ell-1}.$$

One can see this iteration as a first order linear system driven by the signal $p\ell$ and using the convolution integral we have a closed-form expression for F_k

$$F_k = \sum_{\ell=0}^{k} q^{k-\ell} p\ell + q^k C$$

We show below that, equivalently we have

$$F_k = k - \frac{q}{p} + Cq^k + \frac{q}{p}q^k \tag{2}$$

and therefore (1) is equivalent to

$$k - 1 - \frac{q}{p} + Cq^{k-1} + \frac{q}{p}q^{k-1} < k$$

or equivalently

$$q^{k-1} < \frac{1 + q/p}{C + q/p} = \frac{1}{pC + q}$$

This means that the policy is to park if $k \leq i$ where i is the smallest integer such that

$$q^{i-1} < \frac{1}{pC + q}$$

It rests to prove (2):

$$\begin{aligned}
F_k &= \sum_{\ell=0}^{k} q^{k-\ell} p\ell + q^k C \\
&= q^k (p \sum_{\ell=0}^{k} q^{-\ell}\ell) + q^k C \\
&= q^k (p \frac{1}{p} \frac{(k - \frac{q}{p})}{q^k} + \frac{q}{p}) + q^k C \\
&= k - \frac{q}{p} + q^k \frac{q}{p} + q^k C
\end{aligned}$$

where we use the fact that

$$\begin{aligned}
\sum_{\ell=0}^{k} q^{-\ell}\ell &= \sum_{\ell=1}^{k} q^{-\ell} (\sum_{j=1}^{\ell} 1) = \sum_{j=1}^{k} \sum_{\ell=j}^{k} q^{-\ell} \\
&= \sum_{j=1}^{k} \frac{q^{-j} - q^{-(k+1)}}{1 - \frac{1}{q}} = \sum_{j=1}^{k} \frac{q^{-(k)} - q^{-j+1}}{p} \\
&= \frac{kq^{-(k)} - \frac{(1-q^{-k})}{1-\frac{1}{q}}}{p} = \frac{1}{p} \frac{(k - \frac{q}{p})}{q^k} + \frac{q}{p^2}
\end{aligned}$$

Problem 2.6 (solution not provided)

Problem 2.7 The answer is:

- stop immediately if
$$x_0 \in \{5, 6\}$$
- otherwise, stop after one additional throw, if the sum of the two throws is
$$x_1 \in \{5, 6, 7\}$$
- otherwise, stop after two additional throws, if the sum of the three throws is
$$x_2 \in \{5, 6, 7\}$$
- otherwise, stop after three additional throws, if the sum of the four throws is
$$x_3 \in \{5, 6, 7\}$$
- otherwise, stop after four additional throws, if the sum of the five throws is
$$x_4 \in \{5, 6, 7\}$$
- otherwise, stop after five additional throws, if the sum of the six throws is
$$x_5 \in \{6, 7\}$$
- otherwise, stop after six additional throws, when the sum of the seven throws is seven.

Moreover, this is the probability of winning, denoted by p, given the initial throw x_0, if the optimal policy is used

x_0	1	2	3	4	5	6
p	0.3920	0.336	0.288	0.2469	0.3194	0.4707

There are several ways to justify this policy, the one presented here casts the problem in the canonical form of stochastic dynamic programming

$$x_{k+1} = f_k(x_k, u_k, w_k)$$

$$\mathbb{E}[\sum_{k=0}^{h-1} g_k(x_k, u_k, w_k) + g_h(x_h)]$$

To this end let us define the state x_k for seven stages, ($h = 7$, six decision stages) $k \in \{0, 1, 2, 3, 4, 5, 6, 7\}$. The state belongs to different discrete spaces $x_k \in X_k$. For $k = 0$, $X_0 := \{1, 2, 3, 4, 5, 6\}$ and x_k is the initial value of the initial throw of player A. For $k \geq 1$ the discrete state space are $X_k = \{k+1, k+2, \ldots, 10\}$, with the following interpretation:

> If x_k takes values from $k + 1$ to 7 it means that x_k is the sum of points up to stage k of player A
>
> If $x_k = 8$ it means that player A has decided to stop the game at an earlier stage $\leq k - 1$ and has won (after waiting for player B to play).
>
> If $x_k = 9$ it means that player A has decided to stop the game at an earlier stage $\leq k - 1$ and has lost (after waiting for player B to play).
>
> If $x_k = 10$ it means that player A, since it busted.

Let us define also the control input space $U_k(x_k)$ as

$$U_k(x_k) := \begin{cases} \{1,2\} \text{ if } x_k \leq 7 \\ 1 \text{ if } x_k \in \{8,9,10\} \end{cases}$$

The meaning is as follows. The decisions at states less or equal than 7 (correspond to situations where the game is on) are either 'stop' (1 by convention) or 'continue' (2 by convention) and, as we shall see, correspond to different transitions for x_{k+1}. The decisions at states 8,9,10 are simply fictitious decisions to prolong the game so that it has a fixed number of stages as required by the canonical form of DP (so under these decisions the state remains the same $x_{k+1} = x_k$).

Finally, the disturbances. Let us define $w_k = (w_k^1, w_k^2)$, where $w_k^1 \in \{1,2,3,4,5,6\}$ is a random variable indicating which value of the dice happened at stage k and $w_k^2 \in \{1,2\}$ is a random variable indicating if player B starts playing at this stage with a given value of x_k and an initial throw of two if he/she wins ($w_k^2 = 1$) or looses ($w_k^2 = 2$). Note that w_k^1 only needs to be defined if the game is on ($x_k \leq 7$) and w_k^2 only needs to be defined if $u_k = 1$, that is player A decided to stop at stage k. However, we can define these random variable for every state, input and stage and if they are not needed they will not play a role. The probability distribution are also necessary ingredients. The probability distribution of w_k^1 is easy to compute

$$\text{Prob}[w_k^1 = i] = \frac{1}{6}, \quad \text{for } i \in \{1,2,3,4,5,6\}$$

In turn, the probability distribution of w_k^1 is considerably harder and depends on the state x_k (only matters for $x_k \leq 7$ otherwise any probability distribution can be defined since they do not play a role). Before proceeding note that player B is a rational agent and will have therefore a trivial policy: keep on playing until it makes more points than player A (or busts, in which case he/she looses). Define

$$p_{j,i} := \text{Prob}[w_k^2 = 1 | x_k = j \wedge \text{ Player B starts with } i \text{ points (initial throw)} \wedge u_k = 1].$$

Although we are only interested in $p_{j,2}$ defining $p_{j,i}$ for a different i is useful. It is clear that

$$p_{j,i} = 1 \text{ if } j \leq i$$

(in words: if the initial number of points of player B is already larger or equal than those of A when A decided to stop the probability of B winning is one because B will immediately stop). The following recursion holds when $j > i$

$$p_{j,i} = \begin{cases} \frac{1}{6}(7 - j + 1 + \sum_{\ell=i+1}^{j-1} p_{j,\ell}) \text{ if } j > i+1 \\ \frac{7-j+1}{6} \text{ if } j = i+1 \end{cases} .$$

This follows simply be conditioning over the possible outcomes of w_k^1 and taking into account that if Player B obtains a sum of points still less than j it will continue playing (and the probability of winning is the same as if it actually had started with those many points, say $\bar{\imath}$ and player A had j points, that is $p_{j,\bar{\imath}}$):

$$
\begin{aligned}
p_{j,i} &:= \text{Prob}[w_k^1 = 1 | x_k = j \wedge \text{ Player B starts with } i \text{ points (initial throw)} \wedge u_k = 1].\\
&= \sum_{m=1}^{6} \text{Prob}[w_k^1 = 1 | x_k = j \wedge \text{ Player B starts with } i \text{ points } \wedge u_k = 1 \wedge w_k = m]\text{Prob}[w_k = m]\\
&= \sum_{m=1}^{6} \text{Prob}[w_k^1 = 1 | x_k = j \wedge \text{ Player B starts with } i+m \text{ points } \wedge u_k = 1]\frac{1}{6}\\
&= \frac{1}{6}(\underbrace{7-j+1}_{\text{occurences for which} j \le i+m \le 7} + \sum_{m=1}^{j-i-1} \underbrace{\text{Prob}[w_k^1 = 1 | x_k = j \wedge \text{ Player B starts with } i+m \text{ points } \wedge u_k = 1])}_{p_{j,i+m}}\\
&= \begin{cases} \frac{1}{6}(7-j+1+\sum_{\ell=i+1}^{j-1} p_{j,\ell}) \text{ if } j > i+1 \\ \frac{7-j+1}{6} \text{ if } j = i+1 \end{cases} .
\end{aligned}
$$

which for $i \in \{2,3,4,5,6\}$ leads to

$$p_{3,2} = \tfrac{5}{6}$$

$$
\begin{aligned}
p_{4,3} &= \frac{4}{6}\\
p_{4,2} &= \frac{1}{6}(p_{4,3}+4) = \frac{7}{6}\frac{4}{6}
\end{aligned}
$$

$$
\begin{aligned}
p_{5,4} &= \frac{3}{6}\\
p_{5,3} &= \frac{1}{6}(p_{5,4}+3) = \frac{7}{6}\frac{3}{6}\\
p_{5,2} &= \frac{1}{6}(p_{5,3}+p_{5,4}+3) = (\frac{7}{6})^2\frac{3}{6}
\end{aligned}
$$

$$
\begin{aligned}
p_{6,5} &= \frac{2}{6}\\
p_{6,4} &= \frac{1}{6}(p_{6,5}+2) = \frac{7}{6}\frac{2}{6}\\
p_{6,3} &= \frac{1}{6}(p_{6,4}+p_{6,5}+2) = (\frac{7}{6})^2\frac{2}{6}\\
p_{6,2} &= \frac{1}{6}(p_{6,3}+p_{6,4}+p_{6,5}+2) = (\frac{7}{6})^3\frac{2}{6}
\end{aligned}
$$

$$
\begin{aligned}
p_{7,6} &= \frac{1}{6}\\
p_{7,5} &= \frac{1}{6}(p_{7,6}+1) = \frac{7}{6}\frac{1}{6}\\
p_{7,4} &= \frac{1}{6}(p_{7,5}+p_{7,6}+1) = (\frac{7}{6})^2\frac{1}{6}\\
p_{7,3} &= \frac{1}{6}(p_{7,4}+p_{7,5}+p_{7,6}+1) = (\frac{7}{6})^3\frac{1}{6}\\
p_{7,2} &= \frac{1}{6}(p_{7,3}+p_{7,4}+p_{7,5}+p_{7,6}+1) = (\frac{7}{6})^4\frac{1}{6}
\end{aligned}
$$

Therefore, defining $\bar{p}_{j,2} = 1 - p_{j,2}$ (probability that A wins when he/she stops with j points):

$\bar{p}_{1,2}$	$\bar{p}_{2,2}$	$\bar{p}_{3,2}$	$\bar{p}_{4,2}$	$\bar{p}_{5,2}$	$\bar{p}_{6,2}$	$\bar{p}_{7,2}$
0	0	$1-\frac{5}{6}$	$1-(\frac{7}{6})\frac{4}{6}$	$1-(\frac{7}{6})^2\frac{3}{6}$	$1-(\frac{7}{6})^3\frac{2}{6}$	$1-(\frac{7}{6})^4\frac{1}{6}$

or in terms of numerical values

$\bar{p}_{1,2}$	$\bar{p}_{2,2}$	$\bar{p}_{3,2}$	$\bar{p}_{4,2}$	$\bar{p}_{5,2}$	$\bar{p}_{6,2}$	$\bar{p}_{7,2}$
0	0	0.1667	0.2222	0.3194	0.4707	0.6912

Given that the state, input and disturbances are characterized we now provide the functions f_k and g_k. First f_k:

$$x_{k+1} = \underbrace{\begin{cases} x_k + w_k^1 \text{ ("game continues") if } w_k^1 \leq 7 - x_k \wedge u_k = 2 \text{ ("continue") } \wedge x_k \in \{1,2,3,4,5,6,7\} \\ 10 \text{ ("busts") if } w_k^1 > 7 - x_k \wedge u_k = 2 \text{ ("continues") } \wedge x_k \in \{1,2,3,4,5,6,7\} \\ 8 \text{ ("A wins") if } w_k^2 = 2 \wedge u_k = 1 \text{ ("stop") } \wedge x_k \in \{1,2,3,4,5,6,7\} \\ 9 \text{ ("B wins") if } w_k^2 = 1 \wedge u_k = 1 \text{ ("stop") } \wedge x_k \in \{1,2,3,4,5,6,7\} \\ x_k \text{ if } x_k \in \{8,9,10\} \end{cases}}_{:=f_k(x_k,u_k,w_k)}$$

expresses the evolution of the state and specifies the rules of the game. For the cost function a reward for player A (by convention equal to -1 and assumed to be negative since A wants to minimize the cost) is only given if A wins the game otherwise it is zero. Player A wins the game when his/her decision is to stop ($u_k = 1$) and $w_k^2 = 2$ (A wins). Note that the probability distribution of w_k^2 depends on the state x_k as described above. Therefore

$$g_k(x_k, u_k, w_k) = \begin{cases} -1 \text{ if } u_k = 1 \wedge w_k^2 = 2 \\ 0 \text{ otherwise} \end{cases}$$

and $g_7(x_7) = 0$.

Since we have casted the problem in the canonical form of stochastic dynamic programming we can now apply the algorithm

---

Terminal cost $J_7(8) = J_7(9) = J_7(10) = 0$

Stage $k = 6$

$x_6 \in \{8,9,10\}$ $J_6(8) = J_6(9) = J_6(10) = 0$

$x_6 = 7$

$$\begin{aligned} J_6(7) &= \min_{u_k \in \{1,2\}} \mathbb{E}[g_6(x_6,u_6,w_6) + J_7(f_6(x_6,u_6,w_6))|x_6 = 7] \\ &= \min\{\mathbb{E}[g_6(x_6,1,w_6) + \underbrace{J_7(\underbrace{f_6(x_6,1,w_6)}_{8 \text{ or } 9})}_{=0}|x_6 = 7], \mathbb{E}[\underbrace{g_6(x_6,2,w_6)}_{=0} + \underbrace{J_7(\underbrace{f_6(x_6,2,w_6)}_{=10(busts)})}_{=0}|x_6 = 7]\} \\ &= \mathbb{E}[g_6(x_6,1,w_6)|x_6 = 7] = -1\text{Prob}[g_6(x_6,1,w_6) = -1] + 0\text{Prob}[g_6(x_6,1,w_6) = 0] \\ &= -1\text{Prob}[w_k^2 = 2|x_6 = 7] = -1\bar{p}_{7,2} = -0.6912 \end{aligned}$$

(optimal choice is $u_6 =$ 1-stop)

Stage $k = 5$

$x_5 \in \{8,9,10\}$ $J_5(8) = J_5(9) = J_5(10) = 0$

$x_5 = 7$ and $u_5 = 1$ $J_5(7) = -0.6912$ (same reasoning as at stage $k = 6$)

$x_5 = 6$

$$\begin{aligned} J_5(6) &= \min_{u_k \in \{1,2\}} \mathbb{E}[g_5(x_5,u_5,w_5) + J_6(f_5(x_5,u_5,w_5))|x_5 = 6] \\ &= \min\{\mathbb{E}[g_5(x_5,1,w_5) + \underbrace{J_6(\underbrace{f_5(x_5,1,w_5)}_{8 \text{ or } 9})}_{=0}|x_5 = 6], \mathbb{E}[\underbrace{g_5(x_5,2,w_5)}_{=0} + J_6(\underbrace{f_5(x_5,2,w_5))}_{=10(busts) w.prob.5/6, =7 w.prob.1/6}|x_5 = 6]\} \\ &= \min\{\mathbb{E}[g_5(x_5,1,w_5)|x_5 = 6], \frac{1}{6}J_6(7)\} = \\ &= \min\{-1\text{Prob}[w_k^2 = 2|x_5 = 6], \frac{1}{6}J_6(7)\} = \min\{-\bar{p}_{6,2}, -\frac{1}{6}0.6912\} = \min\{-0.4707, -\frac{1}{6}0.6912\} = -0.4707 \end{aligned}$$

(optimal choice is $u_5 = 1$-stop)

Stage $k = 4$

$x_4 \in \{8, 9, 10\}$ $J_4(8) = J_4(9) = J_4(10) = 0$

$x_4 = 7$ $J_4(7) = -0.6912$ and $u_4 = 1$ (same reasoning as at stage $k = 6$)

$x_4 = 6$ $J_4(6) = -0.4707$ and $u_4 = 1$ (same reasoning as at stage $k = 5$)

$x_4 = 5$

$$
\begin{aligned}
J_4(5) &= \min_{u_k \in \{1,2\}} \mathbb{E}[g_4(x_4, u_4, w_4) + J_5(f_4(x_4, u_4, w_4))|x_4 = 5] \\
&= \min\{\mathbb{E}[g_4(x_4, 1, w_4) + \underbrace{J_5(\underbrace{f_4(x_4, 1, w_4))}_{\text{8 or 9}}}_{=0} |x_4 = 5], \mathbb{E}[\underbrace{g_4(x_4, 2, w_4)}_{=0} \\
&+ J_5(\underbrace{f_4(x_4, 2, w_4))}_{=10(busts)w.prob.5/6,=6w.prob.1/6,=7w.prob.1/6} |x_4 = 5]\} \\
&= \min\{\mathbb{E}[g_4(x_4, 1, w_4)|x_4 = 5], \frac{1}{6}J_5(6) + \frac{1}{6}J_5(7)\} = \\
&= \min\{-1\text{Prob}[w_k^2 = 2|x_4 = 5], \frac{1}{6}J_5(6) + \frac{1}{6}J_5(7)\} \\
&= \min\{-\bar{p}_{5,2}, -\frac{1}{6}0.4707 - \frac{1}{6}0.6912\} = \min\{-0.3194, -0.1937\} = -0.3194
\end{aligned}
$$

(optimal choice is $u_4 = 1$-stop)

Stage $k = 3$

$x_3 \in \{8, 9, 10\}$ $J_3(8) = J_3(9) = J_3(10) = 0$

$x_3 = 7$ $J_3(7) = -0.6912$ and $u_3 = 1$ (same reasoning as at stage $k = 6$)

$x_3 = 6$ $J_3(6) = -0.4707$ and $u_3 = 1$ (same reasoning as at stage $k = 5$)

$x_3 = 5$ $J_3(5) = -0.3194$ and $u_3 = 1$ (same reasoning as at stage $k = 4$)

$x_3 = 4$

$$
\begin{aligned}
J_3(4) &= \min_{u_k \in \{1,2\}} \mathbb{E}[g_3(x_3, u_3, w_3) + J_4(f_3(x_3, u_3, w_3))|x_3 = 4] \\
&= \min\{\mathbb{E}[g_3(x_3, 1, w_3) + \underbrace{J_4(\underbrace{f_3(x_3, 1, w_3))}_{\text{8 or 9}}}_{=0} |x_3 = 4], \mathbb{E}[\underbrace{g_3(x_3, 2, w_3)}_{=0} \\
&+ J_4(\underbrace{f_3(x_3, 2, w_3))}_{=10(busts)w.prob.5/6,=5w.prob.1/6,=6w.prob.1/6,=7w.prob1/6} |x_3 = 4]\} \\
&= \min\{\mathbb{E}[g_3(x_3, 1, w_3)|x_3 = 4], \frac{1}{6}J_4(5) + \frac{1}{6}J_4(6) + \frac{1}{6}J_4(7)\} = \\
&= \min\{-1\text{Prob}[w_k^2 = 2|x_3 = 4], \frac{1}{6}J_4(5) + \frac{1}{6}J_4(6) + \frac{1}{6}J_4(7)\} \\
&= \min\{-\bar{p}_{4,2}, -\frac{1}{6}0.3194 - \frac{1}{6}0.4707 - \frac{1}{6}0.6912\} = \min\{-0.2222, -0.2469\} = -0.2469
\end{aligned}
$$

(optimal choice is $u_3 = 2$-continue)

Stage $k = 2$

$x_2 \in \{8, 9, 10\}$ $J_2(8) = J_2(9) = J_2(10) = 0$

$x_2 = 7$ $J_2(7) = -0.6912$ and $u_2 = 1$ (same reasoning as at stage $k = 6$)

$x_2 = 6$ $J_2(6) = -0.4707$ and $u_2 = 1$ (same reasoning as at stage $k = 5$)

$x_2 = 5$ $J_2(5) = -0.3194$ and $u_2 = 1$ (same reasoning as at stage $k = 4$)

$x_2 = 4$ $J_2(4) = -0.2469$ and $u_2 = 2$ (same reasoning as at stage $k = 3$)

$x_2 = 3$

$$\begin{aligned} J_2(3) &= \min\{\underbrace{-\bar{p}_{3,2}}_{\text{stop}}, \underbrace{\frac{1}{6}J_3(4) + \frac{1}{6}J_3(5) + \frac{1}{6}J_3(6) + \frac{1}{6}J_3(7)}_{\text{continue}}\} \\ &= \min\{-0.1667, -0.2880\} = -0.2880 \end{aligned}$$

(optimal choice is $u_2 = 2$-continue)

1

Stage $k = 1$

$x_1 \in \{8, 9, 10\}$ $J_1(8) = J_1(9) = J_1(10) = 0$

$x_1 = 7$ $J_1(7) = -0.6912$ and $u_1 = 1$ (same reasoning as at stage $k = 6$)

$x_1 = 6$ $J_1(6) = -0.4707$ and $u_1 = 1$ (same reasoning as at stage $k = 5$)

$x_1 = 5$ $J_1(5) = -0.3194$ and $u_1 = 1$ (same reasoning as at stage $k = 4$)

$x_1 = 4$ $J_1(4) = -0.2469$ and $u_1 = 2$ (same reasoning as at stage $k = 3$)

$x_1 = 3$ $J_1(3) = -0.2880$ and $u_1 = 2$ (same reasoning as at stage $k = 2$)

$x_1 = 2$

$$\begin{aligned} J_1(2) &= \min\{\underbrace{-\bar{p}_{2,2}}_{\text{stop and loose for sure}}, \underbrace{\frac{1}{6}J_2(3) + \frac{1}{6}J_2(4) + \frac{1}{6}J_2(5) + \frac{1}{6}J_2(6) + \frac{1}{6}J_2(7)}_{\text{continue}}\} \\ &= \min\{0, -0.3360\} = -0.3360 \end{aligned}$$

(optimal choice is $u_1(2) = 2$-continue)

Stage $k = 0$

$x_0 = 6$ $J_0(6) = -0.4707$ and $u_0 = 1$ (same reasoning as at stage $k = 5$)

$x_0 = 5$ $J_0(5) = -0.3194$ and $u_0 = 1$ (same reasoning as at stage $k = 4$)

$x_0 = 4$ $J_0(4) = -0.2469$ and $u_0 = 2$ (same reasoning as at stage $k = 3$)

$x_0 = 3$ $J_0(3) = -0.2880$ and $u_0 = 2$ (same reasoning as at stage $k = 2$)

$x_0 = 2$ $J_0(2) = -0.3360$ and $u_0 = 2$ (same reasoning as at stage $k = 1$)

$x_0 = 1$

$$\begin{aligned} J_0(1) &= \min\{\underbrace{-p_{1,2}}_{\text{stop and loose for sure}}, \underbrace{\frac{1}{6}J_0(2) + \frac{1}{6}J_0(3) + \frac{1}{6}J_0(4) + \frac{1}{6}J_0(5) + \frac{1}{6}J_0(6) + \frac{1}{6}J_0(7)}_{\text{continue}}\} \\ &= \min\{0, -0.3920\} = -0.3920 \end{aligned}$$

(optimal choice is $u_0 = 2$-continue)

3. Search methods for graphs

Problem 3.1

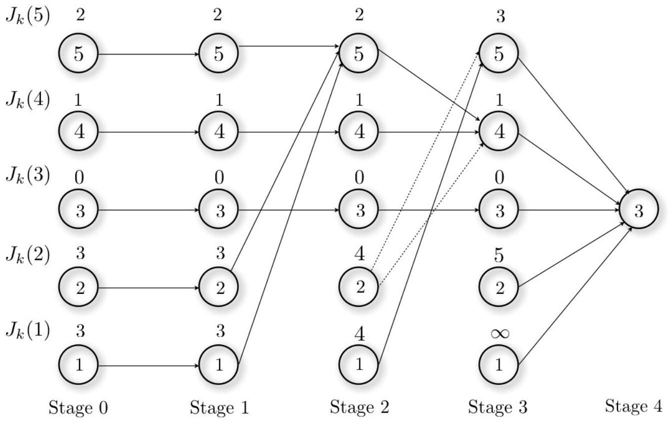


Figure 17: DP algorithm applied to transition diagram associated to a graph

i) $J_k(i)$- cost of the shortest path associated with $4-k$ loops.

Stage 2

State	
1	$\min \begin{bmatrix}\infty & 5 & 0 & 1 & 3\end{bmatrix} + \begin{bmatrix}0 & 0 & \infty & \infty 1\end{bmatrix} = 4$
2	$\min \begin{bmatrix}\infty & 5 & 0 & 1 & 3\end{bmatrix} + \begin{bmatrix}3 & 0 & 5 & 3 & 1\end{bmatrix} = 4$
3	$\min \begin{bmatrix}\infty & 5 & 0 & 1 & 3\end{bmatrix} + \begin{bmatrix}\infty & 5 & 0 & 1 & 3\end{bmatrix} = 0$
4	$\min \begin{bmatrix}\infty & 5 & 0 & 1 & 3\end{bmatrix} + \begin{bmatrix}\infty & \infty & 1 & 0 & 1\end{bmatrix} = 1$
5	$\min \begin{bmatrix}\infty & 5 & 0 & 1 & 3\end{bmatrix} + \begin{bmatrix}1 & 1 & 3 & 1 & 0\end{bmatrix} = 2$

Stage 1

State	
1	$\min \begin{bmatrix}4 & 4 & 0 & 1 & 2\end{bmatrix} + \begin{bmatrix}0 & 0 & \infty & \infty 1\end{bmatrix} = 3$
2	$\min \begin{bmatrix}4 & 4 & 0 & 1 & 2\end{bmatrix} + \begin{bmatrix}3 & 0 & 5 & 3 & 1\end{bmatrix} = 3$
3	$\min \begin{bmatrix}4 & 4 & 0 & 1 & 2\end{bmatrix} + \begin{bmatrix}\infty & 5 & 0 & 1 & 3\end{bmatrix} = 0$
4	$\min \begin{bmatrix}4 & 4 & 0 & 1 & 2\end{bmatrix} + \begin{bmatrix}\infty & \infty & 1 & 0 & 1\end{bmatrix} = 1$
5	$\min \begin{bmatrix}4 & 4 & 0 & 1 & 2\end{bmatrix} + \begin{bmatrix}1 & 1 & 3 & 1 & 0\end{bmatrix} = 2$

Stage 0

State	
1	$\min \begin{bmatrix}3 & 3 & 0 & 1 & 2\end{bmatrix} + \begin{bmatrix}0 & 0 & \infty & \infty 1\end{bmatrix} = 3$
2	$\min \begin{bmatrix}3 & 3 & 0 & 1 & 2\end{bmatrix} + \begin{bmatrix}3 & 0 & 5 & 3 & 1\end{bmatrix} = 3$
3	$\min \begin{bmatrix}3 & 3 & 0 & 1 & 2\end{bmatrix} + \begin{bmatrix}\infty & 5 & 0 & 1 & 3\end{bmatrix} = 0$
4	$\min \begin{bmatrix}3 & 3 & 0 & 1 & 2\end{bmatrix} + \begin{bmatrix}\infty & \infty & 1 & 0 & 1\end{bmatrix} = 1$
5	$\min \begin{bmatrix}3 & 3 & 0 & 1 & 2\end{bmatrix} + \begin{bmatrix}1 & 1 & 3 & 1 & 0\end{bmatrix} = 2$

A shortest path is depicted in Figure 18. Note that both the optimal decisions specified in a optimal policy shown in Figure 17 and the optimal / shortest path is not unique.

ii)

Iteration	Open list	Previous node
0	$\{1,0\}$	
1	$\{2,0\},\{5,1\}$	$\beta(2)=1$, $\beta(5)=1$
2	$\{5,1\},\{3,5\},\{4,3\}$	$\beta(3)=2$, $\beta(4)=2$
3	$\{3,4\},\{4,2\}$	$\beta(4)=5$
4	$\{3,3\}$	$\beta(3)=4$

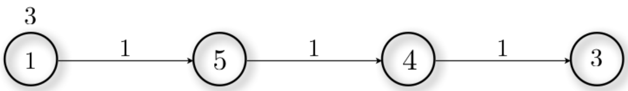


Figure 18: Optimal path

A shortest path can be obtained from this table and is given by 1->5->4->3, as also depicted in Figure (18).

iii)

State	First node of the optimal path to 3
1	5
2	5
3	3
4	3
5	4

Problem 3.2 We start by relabeling the nodes as depicted in Figure 19. Computing the optimal path from node 1 to node 13 we obtain

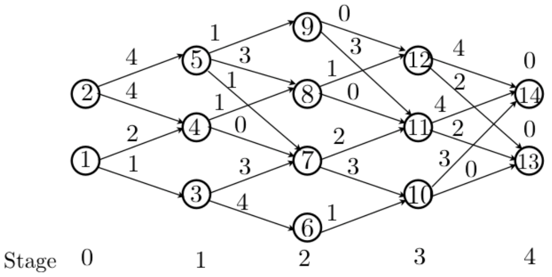


Figure 19: Graph with nodes relabeled

Iteration	List of open nodes (node i,estimate of the distance from i to 2)
0	$\{1, 0\}$
1	$\{3, 1\}, \{4, 2\}$
2	$\{4, 2\}, \{6, 5\}, \{7, 4\}$
3	$\{6, 5\}, \{7, 2\}, \{8, 3\}$
4	$\{6, 5\}, \{8, 3\}, \{10, 5\}, \{11, 4\}$
5	$\{6, 5\}, \{10, 5\}, \{11, 3\}, \{12, 4\}$
6	$\{6, 5\}, \{10, 5\}, \{12, 4\}, \{13, 5\}, \{14, 7\}$
7	$\{6, 5\}, \{13, 5\}, \{14, 7\}$

and we get that the distance is 5. Computing the optimal path from node 1 to node 14 we obtain a distance of 7. Therefore the optimal path from node 1 at stage 0 to the final stage has a cost of 5 and from the table obtained with the Dijkstra's algorithm we can conclude that the optimal path is

Stage	0	1	2	3	4
Node	1	2	3	2	1

Computing the optimal path from node 2 to node 13 we obtain

Iteration	List of open nodes (node i,estimate of the distance from i to 2)
0	$\{2,0\}$
1	$\{4,4\},\{5,4\}$
2	$\{5,4\},\{7,4\},\{8,5\}$
3	$\{7,4\},\{8,5\},\{9,5\}$
4	$\{8,5\},\{9,5\},\{10,7\},\{11,6\}$
5	$\{9,5\},\{10,7\},\{11,5\},\{12,6\}$
6	$\{10,7\},\{11,5\},\{12,5\}$
7	$\{10,7\},\{12,5\},\{13,7\},\{14,9\}$
8	$\{10,7\},\{13,7\},\{14,9\}$
9	$\{13,7\},\{14,9\}$

and we get that the distance is 7. Computing the optimal path from node 2 to node 14 we obtain a distance of 9. Therefore the optimal path from node 2 at stage 0 to the final stage has a cost of 7 and from the table obtained with the Dijkstra's algorithm we can conclude that the optimal path is

Stage	0	1	2	3	4
Node	2	2	3	2	1

Problem 3.3 (i) and (ii) can be answered by first running the DP algorithm for the transition diagram associated with the graph. This results in the costs-to go and decisions depicted in Figure 20. To obtain this policy one could first construct the mentioned transition diagram. However this is tedious, since one has to place arrows connecting all the nodes at every stage to all the nodes of the proceeding stage. As an alternative one can start by placing the cost-to-go associated with the first step of the algorithm (stage 4 in this case), which are simply the costs of the links connecting each node to the destination (or infinity if there is no link). In the example this corresponds to the row in the table associated with Maui Then at the second step (stage 3 in this case) one can compute for each state i

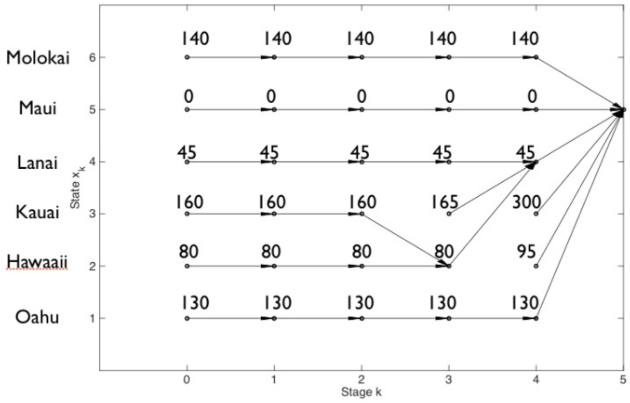


Figure 20: DP algorithm applied to transition diagram associated to a graph

$$\min_j c_{ij} + J_{k+1}(j) \tag{3}$$

where c_{ij} is the cost of reaching node j from i (which can be found in the table) and $J_{k+1}(j)$ is the cost-to-go obtained in the previous step associated with node j. For example for node 1 associated with Oahu, c_{ij} are obtained from the row in the table associated with Oahu

$$\begin{bmatrix}0 & 95 & 75 & 105 & 130 & 90\end{bmatrix}$$

and (3) amounts to computing

$$\min\begin{bmatrix}0 & 95 & 75 & 105 & 130 & 90\end{bmatrix} + \begin{bmatrix}130 & 95 & 300 & 45 & 0 & 140\end{bmatrix}$$

which is 130 and corresponds to picking the next node to be 1 (itself, this means that the next node along the optimal path to the destination did not change). For node 2 we have to compute

$$\min\begin{bmatrix}90 & 0 & 105 & 35 & 95 & 120\end{bmatrix} + \begin{bmatrix}130 & 95 & 300 & 45 & 0 & 140\end{bmatrix}$$

which is 80 and corresponds to picking the next node to be node 4 (Lanai). The algorithm continues to the previous stage, etc.

At each iteration of the algorithm i the cost-to-go associated with a given node i corresponds to the optimal cost to reach the destination node with i hops. As such, looking at the line of costs-to-go in Figure (20) associated with the departure node Kauai we obtain

- With 2 hops the optimal cost is 165 and corresponds to the path Kauai->Lanai->Maui
- With 3 hops the optimal cost is 160 and corresponds to the path Kauai->Hawaii->Lanai->Maui
- With 4 hops the optimal cost is 160 and corresponds to the path Kauai->Hawaii->Lanai->Maui
- With 5 hops the optimal cost is 160 and corresponds to the path Kauai->Hawaii->Lanai->Maui

	Iteration i	List of open nodes (node i,estimate of the distance from i to Kauai)
	0	$\{\text{Kauai}, 0\}$
	1	$\{\text{Oahu}, 70\}, \{\text{Hawaii}, 80\}, \{\text{Lanai}, 120\}, \{\text{Maui}, 300\}, \{\text{Molokai}, 80\}$
(iii)	2	$\{\text{Hawaii}, 80\}, \{\text{Lanai}, 120\}, \{\text{Maui}, 200\}, \{\text{Molokai}, 80\}$
	3	$\{\text{Lanai}, 115\}, \{\text{Maui}, 175\}, \{\text{Molokai}, 80\}$
	4	$\{\text{Lanai}, 115\}, \{\text{Maui}, 175\}$
	5	$\{\text{Maui}, 160\}$

and from the table obtained with the Dijkstra's algorithm we can conclude that the optimal path is Kauai->Hawaii->Lanai->Maui

Problem 3.4

(i) Dijkstra's algorithm

Iteration i	List of open nodes (node i,estimate of the distance from i to node 1)
0	$\{1, 0\}$
1	$\{3, 1\}, \{4, 10\}, \{2, 10\}$
2	$\{6, 2\}, \{4, 10\}, \{2, 10\}$
3	$\{9, 3\}, \{4, 10\}, \{2, 10\}$
4	$\{11, 103\}, \{4, 10\}, \{2, 10\}$
5	$\{11, 103\}, \{7, 20\}, \{2, 10\}$
6	$\{11, 103\}, \{7, 20\}, \{5, 20\}$
7	$\{11, 103\}, \{10, 30\}, \{5, 20\}$
8	$\{11, 103\}, \{10, 30\}, \{8, 30\}$
9	$\{11, 31\}, \{8, 30\}$
10	$\{11, 31\}$

(ii) A* algorithm First apply a transformation $\bar{c}_{ij} = c_{ij} + h(j) - h(i)$, where c_{ij} is the cost from node i to node j, obtaining the transition diagram in Figure 21 (this is only valid because after the transformation $\bar{c}_{ij} \geq 0$).

Iteration i	List of open nodes (node i,estimate of the distance from i to node 1)
0	$\{1, 0\}$
1	$\{4, 25\}, \{1, 51\}, \{2, 35\}$
2	$\{7, 29\}, \{1, 51\}, \{2, 35\}$
3	$\{10, 30\}, \{1, 51\}, \{2, 35\}$
4	$\{11, 31\}$

Problem 3.5

(i) The costs-to-go obtaining with the DP algorithm for the transition diagram associated with the graph (see slide 15 lecture 2) are provided in Figure 22 where the optimal paths from the initial node 2 to the final node 1 are also indicated.

(ii) Shortest path 2->4->3->1 in (i) has length 3, whereas Path 2->4->3->2->4->3->1 has length 1, so the path given by DP does not provide the optimal path.

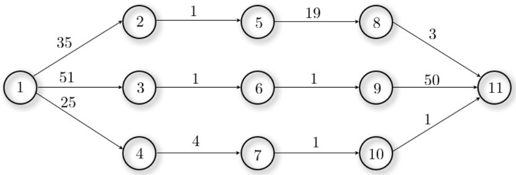


Figure 21: New graph after transformation $\bar{c}_{ij} = c_{ij} + h(j) - h(i)$

(iii) No it is not possible.

(iv) There cannot exist cycles for which the sum of the weights of the links is negative.

(v) Dijkstra's algorithm only works for problems where the weights are non-negative (in this case we could only apply it to (a)).

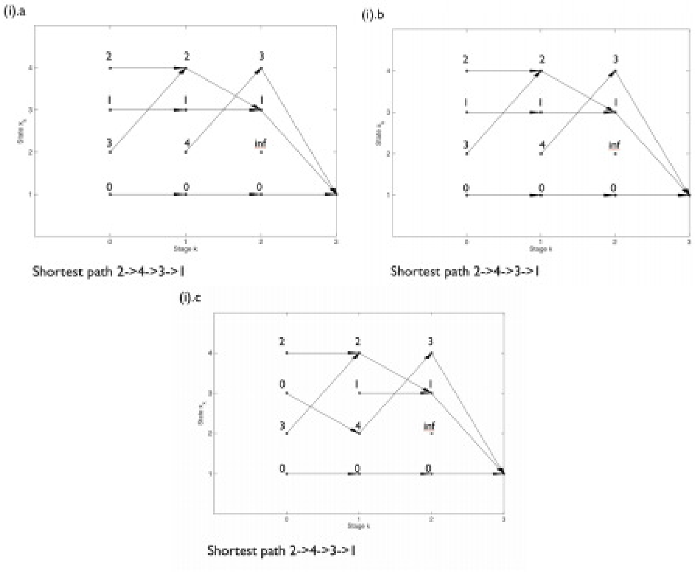


Figure 22: DP

4. Bayes filter and POMDP

Problem 4.1

We can solve this exercise by recursively applying the Bayes rule (correction/update step) and propagating model uncertainty (prediction step). We will do this for the first step to exemplify this process. A more expedite (equivalent) solution process, is to find the matrices P and R, defined in Lecture 4 (slide 12) and iterate the equations of slide 15, lecture 4, which we then discuss.

Let us start by computing the first correction/update step $\text{Prob}[\theta_0|y_0 = \text{green}]$. From Bayes' rule

$$\text{Prob}[\theta_0 = 0|y_0 = \text{green}] = \alpha\text{Prob}[y_0 = \text{green}|\theta_0 = 0]\text{Prob}[\theta_0 = 0] = \alpha 0.8 \times 0.25$$

$$\text{Prob}[\theta_0 = \frac{\pi}{2}|y_0 = \text{green}] = \alpha\text{Prob}[y_0 = \text{green}|\theta_0 = \frac{\pi}{2}]\text{Prob}[\theta = \frac{\pi}{2}] = \alpha 0.1 \times 0.25$$

$$\text{Prob}[\theta_0 = \pi|y_0 = \text{green}] = \alpha\text{Prob}[y_0 = \text{green}|\theta_0 = \pi]\text{Prob}[\theta_0 = \pi] = \alpha 0 \times 0.25$$

$$\text{Prob}[\theta_0 = \frac{3\pi}{2}|y_0 = \text{green}] = \alpha\text{Prob}[y_0 = \text{green}|\theta_0 = \frac{3\pi}{2}]\text{Prob}[\theta_0 = \frac{3\pi}{2}] = \alpha 0.1 \times 0.25$$

where

$$\alpha = 1/(0.8 \times 0.25 + 0.1 \times 0.25 + 0 \times 0.25 + 0.1 \times 0.25)$$

$$\text{Therefore Prob}[\theta_0 = \beta | y_0 = \text{green}] = \begin{cases} 0.8 \text{ if } \alpha = 0 \\ 0.1 \text{ if } \alpha = \dfrac{\pi}{2} \\ 0 \text{ if } \alpha = \pi \\ 0.1 \text{ if } \alpha = \dfrac{3\pi}{2} \end{cases}$$

Let us now apply a prediction step computing Prob$[\theta_1 | y_0 = \text{green}]$

We have, for $\theta_1 = 0$,

$$\begin{aligned} &\text{Prob}[\theta_1 = 0 | y_0 = \text{green}] \\ &= \text{Prob}[\theta_0 + w_0 = 0 | y_0 = \text{green}] \\ &= \sum_{c \in \{0, \pi/2, \pi\}} \text{Prob}[\theta_0 = -c | y_0 = \text{green}]\text{Prob}[w_0 = c] \\ &= \text{Prob}[\theta_0 = 0 | y_0 = \text{green}]\text{Prob}[w_0 = 0] + \text{Prob}[\theta_0 = -\pi/2 | y_0 = \text{green}]\text{Prob}[w_0 = \pi/2] = 0.8 \times 0.2 + 0.1 \times 0.6 = 0.22 \end{aligned}$$

for $\theta_1 = \pi/2$,

$$\begin{aligned} &\text{Prob}[\theta_1 = \pi/2 | y_0 = \text{green}] \\ &= \text{Prob}[\theta_0 + w_0 = \pi/2 | y_0 = \text{green}] \\ &= \sum_{c \in \{0, \pi/2, \pi\}} \text{Prob}[\theta_0 = -c + \pi/2 | y_0 = \text{green}]\text{Prob}[w_0 = c] \\ &= 0.1 \times 0.2 + 0.8 \times 0.6 + 0.1 \times 0.2 = 0.52 \end{aligned}$$

for $\theta_1 = \pi$,

$$\begin{aligned} &\text{Prob}[\theta_1 = \pi | y_0 = \text{green}] \\ &= \text{Prob}[\theta_0 + w_0 = \pi | y_0 = \text{green}] \\ &= \sum_{c \in \{0, \pi/2, \pi\}} \text{Prob}[\theta_0 = -c + \pi | y_0 = \text{green}]\text{Prob}[w_0 = c] \\ &= 0 \times 0.2 + 0.1 \times 0.6 + 0.8 \times 0.2 = 0.22 \end{aligned}$$

for $\theta_1 = 3\pi/2$,

$$\begin{aligned} &\text{Prob}[\theta_1 = 3\pi/2 | y_0 = \text{green}] \\ &= \text{Prob}[\theta_0 + w_0 = 3\pi/2 | y_0 = \text{green}] \\ &= \sum_{c \in \{0, \pi/2, \pi\}} \text{Prob}[\theta_0 = -c + 3\pi/2 | y_0 = \text{green}]\text{Prob}[w_0 = c]\cdot \\ &= 0.1 \times 0.2 + 0.1 \times 0.2 = 0.02 \end{aligned}$$

It is clear that this process is time-consuming. Let us organize the information better to make the process more expedite.

Define a 4×4 matrix P such that the entry P_{ij} has the following meaning

$$P_{ij} = \text{Prob}[\theta_{k+1} = (i-1)\frac{\pi}{2} | \theta_k = (j-1)\frac{\pi}{2}]$$

We then have

$$P = \begin{bmatrix} 0.2 & 0 & 0.2 & 0.6 \\ 0.6 & 0.2 & 0 & 0.2 \\ 0.2 & 0.6 & 0.2 & 0 \\ 0 & 0.2 & 0.6 & 0.2 \end{bmatrix}$$

Define also a matrix R such that the entry R_{ij} has the following meaning

$$R_{ij} = \text{Prob}[y_k = f(i) | \theta_k \theta_k = (j-1)\frac{\pi}{2}]$$

where $f(1) = \text{green}$, $f(2) = \text{orange}$, $f(3) = \text{red}$, $f(4) = \text{blue}$. Then

$$R = \begin{bmatrix} 0.8 & 0.1 & 0 & 0.1 \\ 0.1 & 0.8 & 0.1 & 0 \\ 0 & 0.1 & 0.8 & 0.1 \\ 0.1 & 0 & 0.1 & 0.8 \end{bmatrix}$$

Then the desired probability distributions $p_k = \begin{bmatrix} \text{Prob}[\theta_k = 0|I_k] \\ \text{Prob}[\theta_k = \pi|I_k] \\ \text{Prob}[\theta_k = \pi/2|I_k] \\ \text{Prob}[\theta_k = 3\pi/2|I_k] \end{bmatrix}$ are obtained by iterating the recursion, for $k \in \{0, 1, \dots, 4\}$

$$\begin{aligned} q_{k+1} &= P p_k \\ \bar{p}_{k+1} &= D(y_{k+1}) q_{k+1} \\ p_{k+1} &= \frac{\bar{p}_{k+1}}{1^\intercal \bar{p}_{k+1}} \end{aligned}$$

where

$$D(y_{k+1}) = \begin{bmatrix} R_{f^{-1}(y_{k+1})1} & 0 & 0 & 0 \\ 0 & R_{f^{-1}(y_{k+1})2} & 0 & 0 \\ 0 & 0 & R_{f^{-1}(y_{k+1})3} & 0 \\ 0 & 0 & 0 & R_{f^{-1}(y_{k+1})4} \end{bmatrix}$$

with $p_0 = \frac{q_0}{1^\intercal q_0}$, $q_0 = D(y_0)\tilde{p}_0$, where

$$\tilde{p}_0 = \begin{bmatrix} \text{Prob}[\theta_0 = 0] \\ \text{Prob}[\theta_0 = \pi/2] \\ \text{Prob}[\theta_0 = \pi] \\ \text{Prob}[\theta_0 = 3\pi/2] \end{bmatrix}$$

We have then $\tilde{p}_0 = \begin{bmatrix} 1/4 & 1/4 & 1/4 & 1/4 \end{bmatrix}^\intercal$,

$$p_0 = \frac{1}{1/4} \begin{bmatrix} 0.8 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix} \begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix} = \begin{bmatrix} 0.8 \\ 0.1 \\ 0 \\ 0.1 \end{bmatrix}$$

Moreover,

$$q_1 = \begin{bmatrix} 0.2 & 0 & 0.2 & 0.6 \\ 0.6 & 0.2 & 0 & 0.2 \\ 0.2 & 0.6 & 0.2 & 0 \\ 0 & 0.2 & 0.6 & 0.2 \end{bmatrix} \begin{bmatrix} 0.8 \\ 0.1 \\ 0 \\ 0.1 \end{bmatrix} = \begin{bmatrix} 0.22 \\ 0.52 \\ 0.22 \\ 0.04 \end{bmatrix}$$

and

$$p_1 = \frac{1}{0.46} \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.8 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0.22 \\ 0.52 \\ 0.22 \\ 0.04 \end{bmatrix} = \begin{bmatrix} 0.0478 \\ 0.9043 \\ 0.0478 \\ 0 \end{bmatrix}$$

This process, which involves simple matrix multiplications can be continued to obtain the desired probability distributions of the state given the available information resulting in

$$p_2 = \begin{bmatrix} 0.0085 \\ 0.7427 \\ 0.2488 \\ 0 \end{bmatrix}, \quad p_3 = \begin{bmatrix} 0.0176 \\ 0 \\ 0.1696 \\ 0.8129 \end{bmatrix} \quad p_4 = \begin{bmatrix} 0.1962 \\ 0 \\ 0.0140 \\ 0.7899 \end{bmatrix}$$

Problem 4.2 This exercise can be solved as 4.1, taking into account another sensor model. Namely, let us define a 2×4 matrix R such that the entry R_{ij} has the following meaning

$$R_{ij} = \text{Prob}[y_k = i - 1|\theta_k = (j-1)\frac{\pi}{2}]$$

Then

$$R = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

. The matrix P is defined as in 4.1. Only the solution is provided

$$p_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad p_1 = \begin{bmatrix} 0 \\ 0.75 \\ 0.25 \\ 0 \end{bmatrix} \quad p_2 = \begin{bmatrix} 0 \\ 0.1579 \\ 0.5263 \\ 0.3158 \end{bmatrix} \quad p_3 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad p_4 = \begin{bmatrix} 0 \\ 0.75 \\ 0.25 \\ 0 \end{bmatrix}$$

where p_k is defined as in 4.1.

Problem 4.3 (Solution not provided)

Problem 4.4 (Solution not provided)