More Examples on
Probabilistic Dynamic Programming

The characteristic of probabilistic DP is that the states and the immediate returns are probabilistic rather than deterministic.

1 A Variation of the Russian Roulette

In this problem we spin a wheel which is marked with 1 to n. The probability that the wheel stops at the number i is p_i. You have to pay $\$x$ to spin the wheel *up to* m times. The payoff is $2\times$ the number produced in the *last* spin. The problem is to find an optimal strategy for this game.

We want to set this up as a probabilistic DP. We note that

1. We have m spins of the wheel so the natural candidate for the stage variable is: stage $i =$ the i-th spin of the wheel where $i = 1, 2, \cdots, m$.
2. The decision at each stage is either to spin the wheel one more time or end the game right there.
3. The states of stage i is the number obtained in the *last* spin.

Thus we have the following probabilistic DP:

OVF: Let $f_i(j) =$ the maximum expected return given that the game is at stage i and that j is the outcome of the last spin. Thus

$$\begin{pmatrix} \text{Expected payoff at stage } i \\ \text{given } j \text{ is the state} \end{pmatrix} = \begin{cases} 2j & \text{if the game ends,} \\ \sum_{k=1}^{n} p_k f_{i+1}(k) & \text{if the game continues.} \end{cases}$$

Recursive relation: This is given by

$$f_i(j) = \max\left\{ 2j, \sum_{k=1}^{n} p_k f_{i+1}(k) \right\}$$

for $i = 1, 2, 3, \cdots, m$.
OPF: We have

$$P_i(j) = \begin{cases} \text{stop} & \text{if } 2j \ge \sum_{k=1}^{n} p_k f_{i+1}(k) \\ \text{spin once more} & \text{otherwise.} \end{cases}$$

Boundary conditions: We have $f_{m+1}(j) = 2j$ for any j.
Answer: It is $f_1(0)$.

We now illustrate with a numerical example. Suppose $n = 5$ and

i	p_i
1	0.30
2	0.25
3	0.20
4	0.15
5	0.10

And you pay \$5 for up to $m = 4$ spins.

Stage 5 For this stage $f_5(j) = 2j$. So

Spin 4 outcome	Optimum solution	
j	$f_5(j)$	Decision
1	2	End
2	4	End
3	6	End
4	8	End
5	10	End

The decision is easy since the game must end here.

Stage 4 For this stage

$$
\begin{aligned}
f_4(j) &= \max\{2j, \sum_{k=1}^{5} p_k F_5(k)\} \\
&= \max\{2j, 0.3 \times 2 + 0.25 \times 4 + 0.2 \times 6 + 0.15 \times 8 + 0.1 \times 10\} \\
&= \max\{2j, 5\}.
\end{aligned}
$$

So

Spin 3 outcome	Expected Return		Optimum solution	
j	End	Spin	$f_4(j)$	Decision
1	2	5	5	Spin
2	4	5	5	Spin
3	6	5	6	End
4	8	5	8	End
5	10	5	10	End

Stage 3 For this stage

$$
\begin{aligned}
f_3(j) &= \max\{2j, \sum_{k=1}^{5} p_k F_5(k)\} \\
&= \max\{2j, 0.3 \times 5 + 0.25 \times 5 + 0.2 \times 6 + 0.15 \times 8 + 0.1 \times 10\} \\
&= \max\{2j, 6.15\}.
\end{aligned}
$$

So

Spin 3 outcome	Expected Return		Optimum solution	
j	End	Spin	$f_3(j)$	Decision
1	2	6.15	6.15	Spin
2	4	6.15	6.15	Spin
3	6	6.15	6.15	Spin
4	8	6.15	8	End
5	10	6.15	10	End

Stage 2 For this stage

$$\begin{aligned} f_3(j) &= \max\{2j, \sum_{k=1}^{5} p_k F_5(k)\} \\ &= \max\{2j, 0.3 \times 6.15 + 0.25 \times 6.15 + 0.2 \times 6.15 + 0.15 \times 8 + 0.1 \times 10\} \\ &= \max\{2j, 6.8125\}. \end{aligned}$$

So

Spin 3 outcome	Expected Return		Optimum solution	
j	End	Spin	$f_3(j)$	Decision
1	2	6.8125	6.8125	Spin
2	4	6.8125	6.8125	Spin
3	6	6.8125	6.8125	Spin
4	8	6.8125	8	End
5	10	6.8125	10	End

Stage 1 We have

$$\begin{aligned} f_1(0) &= \sum_{k=1}^{5} p_k f_2(k) \\ &= 0.3 \times 6.8125 + 0.25 \times 6.8125 + 0.2 \times 6.8125 + 0.15 \times 8 + 0.1 \times 10 \\ &= 7.3094 \end{aligned}$$

Thus the maximal expected return is $7.3094 - 5 = 2.3094$ and the optimal strategy is given by

Spin No.	What to do
1	Spin
2	If spin 1 = 1, 2, 3 spin again
	If spin 1 = 4, 5 end the game
3	If spin 2 = 1, 2, 3 spin again
	If spin 2 = 4, 5 end the game
4	If spin 3 = 1, 2 spin again
	If spin 3 = 3, 4, 5 end the game

2 Another Game of Chance

In this example you have \$2 and you have to play a game of chance 4 times. The chance of winning a single bet is 0.4 and thus the chance of losing it is 0.6. If you have b dollars then you can bet $0, 1, \cdots, b$ dollars. The goal is to find a strategy that maximizes *your chance* of ending up with at least \$6.

There are some natural elements of DP

1. The stage i is just the i-th bet of the game.
2. The decision at stage i is y_i = how much to bet.
3. The states at stage i is x_i = the money available for betting.

The probabilistic DP is formulated as

OVF: Since our goal is the maximize the chance of ending up with at least \$6, we define $f_i(x_i)$ = the probability that you will have at least \$6 at the end of the 4-th bet given that you have x_i dollars before you make the i-th bet.
Recursion relation: Note that if you have $\$x_i$ at the beginning of stage i and bet $\$y_i$ then the expected return is

$$0.4 \times f_{i+1}(x_i + y_i) + 0.6 \times f_{i+1}(x_i - y_i).$$

Therefore the recursive equation is

$$f_i(x_i) = \max_{0 \le y_i \le x_i} \{0.4 \times f_{i+1}(x_i + y_i) + 0.6 \times f_{i+1}(x_i - y_i\}.$$

OPF: The optimal policy is $P_i(x_i)$ = the y_i the maximizes the last equation.
Boundary Condition: We have

$$f_5(x_5) = \begin{cases} 1 & \text{if } x_5 \ge 6, \\ 0 & \text{if } x_5 < 6. \end{cases}$$

Answer: $f_1(2)$.
The computation goes as follows.

Stage 4.

	y_4								
x_4	0	1	2	3	4	5	6	$f_4(x_4)$	y_4^*
0	0	-	-	-	-	-	-	0	0
1	0	0	-	-	-	-	-	0	0
2	0	0	0	-	-	-	-	0	0
3	0	0	0	0.4	-	-	-	0.4	3
4	0	0	0.4	0.4	0.4	-	-	0.4	2,3, 4
5	0	0.4	0.4	0.4	0.4	0.4	-	0.4	1,2,3, 4,5
≥ 6	1	0.4	0.4	0.4	0.4	0.4	0.4	1	0

Stage 3.

x_3	y_3 = 0	1	2	3	4	5	6	$f_3(x_3)$	y_3^*
0	0	-	-	-	-	-	-	0	0
1	0	0	-	-	-	-	-	0	0
2	0	0.16	0.16	-	-	-	-	0.16	1,2
3	0.4	0.16	0.16	0.4	-	-	-	0.4	0, 3
4	0.4	0.4	0.4	0.4	0.4	-	-	0.4	0, 1, 2,3, 4
5	0.4	0.64	0.64	0.4	0.4	0.4	-	0.64	1,2
≥ 6	1	-	-	-	-	-	-	1	0

Stage 2.

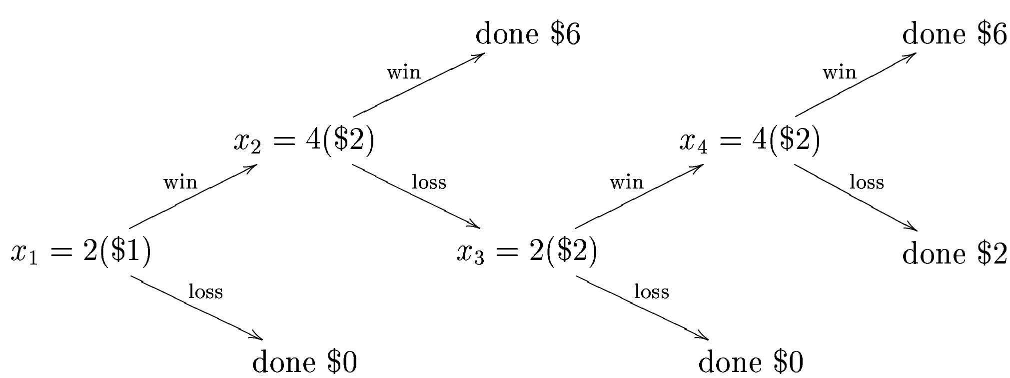
x_2	y_2 = 0	1	2	3	4	5	6	$f_2(x_2)$	y_2^*
0	0	-	-	-	-	-	-	0	0
1	0	0.064	-	-	-	-	-	0.064	1
2	0.16	0.16	0.16	-	-	-	-	0.16	0, 1,2
3	0.4	0.256	0.256	0.4	-	-	-	0.4	0, 3
4	0.4	0.496	0.496	0.4	0.4	-	-	0.496	1, 2
5	0.64	0.64	0.64	0.496	0.4	0.4	-	0.64	0,1
≥ 6	1	-	-	-	-	-	-	1	0

Stage 1.

x_1	y_1 = 0	1	2	$f_1(x_1)$	$y_1 2^*$
2	0.16	0.1984	0.1984	0.1984	1,2

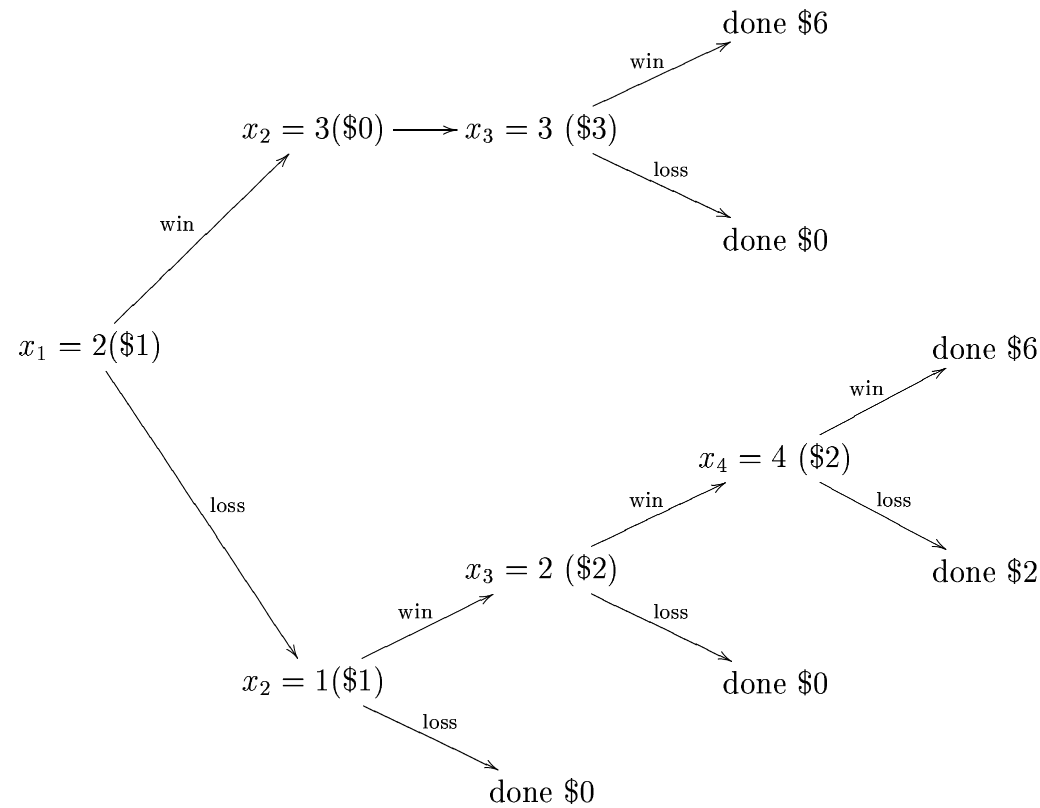
Thus your optimal probability of reaching \$6 is 0.1984. The optimal betting strategies are:

Strategy 1:



The amount inside the bracket is the amount of the bet.

Strategy 2:



3 The Investment Problem

The stochastic version of the investment problem involves stochastic return of the investments represented by a probability distribution. Consider the following problem. You want to invest up to $\$C$ in the stock market over the next n years. The plan is to buy the stock at the beginning of the year and sell it at end of the year. Accumulated money can be reinvested. The degree of risk in the investment is represented by a probability distribution on the return of the stock. There are m different market conditions. Condition i yields a return r_i with probability p_i. The problem is to determine an investment policy to realize the highest accumulation at the end of year n.

Let

$$x_i = \text{money available at the start of year } i$$
$$y_i = \text{money invested at the start of year } i$$

Then the natural ingredients for the DP is

$$\text{Stage } i = \text{Year } i$$
$$\text{States at stage } i = x_i$$
$$\text{Decisions at stage } i = y_i$$

We formulate the DP below.

OVF: Let $f_i(x_i)$ = the maximal expected return for years $i, i+1, \cdots, n$ given x_i dollars available at the start of year i.

Recursion relation: For market condition k, we have

$$x_{i+1} = (1 + r_k)y_i + (x_i - y_i) = r_k y_i + x_i, \quad k = 1, \cdots, m.$$

Therefore the recursion relation is given by

$$f_i(x_i) = \max_{0 \le y_i \le x_i} \left\{ \sum_{k=1}^{m} p_k f_{i+1}(r_k y_i + x_i) \right\}.$$

OPF: This is given by $P_i(x_i)$ = the y_i that maximizes the last equation.

Boundary conditions: We have $f_{n+1}(x_{n+1}) = x_{n+1}$ because no investment occurs after year n. In this case we can also compute

$$\begin{aligned} f_n(x_n) &= \max_{0 \le y_n \le x_m} \left\{ \sum_{k=1}^{m} p_k f_{n+1}(x_n + r_k y_n) \right\} \\ &= \sum_{k=1}^{m} p_k (x_n + r_k y_n) \\ &= \sum_{k=1}^{m} p_k (x_n + r_k x_n) \\ &= x_n \sum_{k=1}^{m} p_k (1 + r_k) \\ &= x_n (1 + p_1 r_1 + \cdots + p_m r_m) \end{aligned}$$

Answer: It is $f_1(C)$.

We give a numerical example to illustrate. Suppose $C = 10,000$ and $n = 4$, $m = 3$ with

i	p_i	r_i
1	0.4	2
2	0.2	0
3	0.4	-1

The computation goes as follows.

Stage 4. In this stage we gave

$$f_4(x_4) = x_4(1 + 0.4 \times 2 + 0.2 \times 0 + 0.4 \times -1) = 1.4x_4.$$

State	$f_4(x_4)$	y_4^*
x_4	$1.4x_4$	$1.4x_4$

Stage 3. In this stage we gave

$$\begin{aligned} f_3(x_3) &= \max_{0 \le y_3 \le x_3} \left\{ \sum_{k=1}^{3} p_k f_4(x_3 + r_k y_3) \right\} \\ &= \max_{0 \le y_3 \le x_3} \{0.4 \times 1.4(x_3 + 2y_3) + 0.2 \times 1.4(x_3 + 0 \times y_3) + 0.4 \times 1.4(x_3 - y_3)\} \\ &= \max_{0 \le y_3 \le x_3} \{1.4x_3 + 0.56y_3\} \\ &= 1.96x_3 \end{aligned}$$

because the function is linear.

State	$f_3(x_3)$	y_3^*
x_3	$1.96x_3$	$1.96x_3$

Stage 2. In this stage we gave

$$\begin{aligned} f_2(x_2) &= \max_{0 \le y_2 \le x_2} \left\{ \sum_{k=1}^{3} p_k f_3(x_2 + r_k y_2) \right\} \\ &= \max_{0 \le y_2 \le x_2} \{0.4 \times 1.96(x_2 + 2y_2) + 0.2 \times 1.96(x_2 + 0 \times y_2) + 0.4 \times 1.96(x_2 - y_2)\} \\ &= \max_{0 \le y_2 \le x_2} \{1.96x_2 + 0.785y_2\} \\ &= 2.744x_2 \end{aligned}$$

because the function is linear.

State	$f_2(x_2)$	y_2^*
x_2	$2.744x_2$	$2.744x_2$

Stage 1. In this stage we gave

$$\begin{aligned} f_1(x_1) &= \max_{0 \le y_1 \le x_1} \left\{ \sum_{k=1}^{3} p_k f_2(x_1 + r_k y_1) \right\} \\ &= \max_{0 \le y_1 \le x_1} \{0.4 \times 2.744(x_1 + 2y_1) + 0.2 \times 2.744(x_1 + 0 \times y_1) + 0.4 \times 2.744(x_1 - y_1)\} \\ &= \max_{0 \le y_1 \le x_1} \{2.744x_1 + 1.0976y_1\} \\ &= 3.8416x_1 \end{aligned}$$

because the function is linear.

State	$f_1(x_1)$	y_1^*
10000	38416	10000

This shows that the optimal policy is to invest all the money at the beginning of each year. The expected accumulated sum in 4 years is $38416.

4 A Variation of the Investment Problem

A supermarket chain purchases 6 cases of milk from a dairy farm for \$100 per case. Each case is sold in the chain's 3 stores for \$200 per case. The dairy must buy back for \$50 per case any milk that is left at the end of the day. The demands at the 3 stores are uncertain and are shown in the following table.

	Daily Demand	Probability
Store 1	1	0.6
	2	0.0
	3	0.4
Store 2	1	0.5
	2	0.1
	3	0.4
Store 3	1	0.4
	2	0.3
	3	0.3

The problem is to determine the optimal policy in allocating the 6 cases of milk to the 3 stores so as to maximize the expected net daily profit.

Since the purchasing cost is always \$600, we will concentrate our attention on maximizing the expected revenue.

Then the natural ingredients for the DP is

$$\text{Stage } i = \text{Store } i$$
$$\text{States at stage } i = \text{No. of cases available } = x_i$$
$$\text{Decisions at stage } i = \text{No. of cases assigned to Store } i = g_i$$

The formulation of DP is as follows.

OVF: Let $r_i(g_i)$ = the expected revenue earned from assigning g_i cases to store i and $f_i(x_i)$ be the maximal expected revenue earned from assigning x_i cases to Stores $i, \cdots, 3$.

Recursion relation: This is given by

$$f_i(x_i) = \max_{0 \le g_i \le x_i} \{r_i(g_i) + f_{i+1}(x_i - g_i)\}.$$

OPF: $P_i(x_i)$ = the y_i the maximizes the above equation.

Boundary conditions: We have $f_4(x_4) = 0$ for all x_4.

Answer: $f_1(6)$.

To solve our problem we first compute the $r_i(g_i)$. Since the maximum demand at each store is 3, we have $0 \le g_3 \le 3$. We illustrate the computation by computing $r_3(2)$. If the demand is ≥ 2 then both cases will be sold and the revenue is \$400. If the demand is 1 then 1 case will be sold and the remaining case will be return for \$50 for a total revenue of \$250. Thus the expected revenue is

$$r_3(20 = 0.4 \times 250 + 0.6 \times 400 = 340.$$

The other $r_i(g_i)$ are computed similarly and we have

g_i	$r_1(g_1)$	$r_2(g_2)$	$r_3(g_3)$
0	0	0	0
1	200	200	200
2	310	325	340
3	420	435	435

Stage 3. In this case $f_3(x_3) = r_3(x_3)$.

	$r_3(g_3)$					
x_3	0	1	2	3	$f_3(x_3)$	g_3^*
0	0	-	-	-	0	0
1	0	200	-	-	200	1
2	0	200	340	-	340	2
3	0	200	340	435	435	3
4	0	200	340	435	435	3
5	0	200	340	435	435	3
6	0	200	340	435	435	3

Stage 2. In this case

$$f_2(x_2) = \max_{0 \le g_2 \le x_2} \{r_2(g_2) + f_3(x_2 - g_2)\}.$$

	$r_2(g_2) + f_3(x_2 - g_2)$					
x_2	0	1	2	3	$f_2(x_2)$	g_2^*
0	0	-	-	-	0	0
1	200	200	-	-	200	0,1
2	340	400	325	-	400	1
3	435	540	525	435	540	1
4	435	635	665	635	665	2
5	435	635	760	775	775	3
6	435	635	760	870	870	3

Stage 1. In this case

$$f_1(x_2) = \max_{0 \le g_1 \le x_1} \{r_1(g_1) + f_2(x_1 - g_1)\}.$$

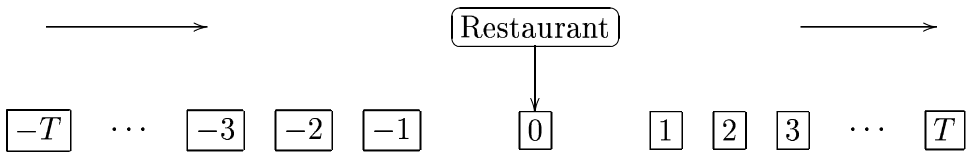
	$r_1(g_1) + f_2(x_1 - g_1)$					
x_1	0	1	2	3	$f_1(x_1)$	g_1^*
6	870	975	975	960	975	1, 2

Thus the optimal policy in allocating the milk is

Store 1	Store 2	Store 3
1	3	2
	or	
2	2	2

5 Parking Problem

You are trying to find a parking space near your favorite restaurant. The situation is illustrated by the following diagram.



When you arrive at the parking space you must decide to take it or not. Once it is passed you cannot return. The probability that the space t is empty is p_t for $t = -T, \cdots, -1, 0, 1, \cdots, T$. If you do not find a parking space, you are embarrassed and it incurs a cost $M \gg 0$. If you park at space t the cost is $|t|$. The problem is to find an optimal parking policy.

If you are at space T the problem is easy: park at T if it is empty; otherwise incur a cost of M. You can then use backward recursion to solve the problem. The natural ingredients for a DP are:

$$\begin{aligned} \text{Stage } i &= \text{Space } i \\ \text{Decisions at } i &= \text{park or not park at } i \\ \text{States at } i &= \text{the parking space is empty or occupied} \end{aligned}$$

We use the convention that if the parking space is occupied then $x_i = 0$ and if it is empty then $x_i = 1$.

OVF: Let $f_i(x_i)$ = the minimum expected cost at space i given the space is in the state x_i.

Recursion relation: It is given by

$$f_i(0) = p_{i+1} f_{i+1}(1) + (1 - p_{i+1}) f_{i+1}(0)$$

$$f_i(1) = \min \begin{cases} |i| & \text{take the space} \\ p_{i+1} f_{i+1}(1) + (1 - p_{i+1}) f_{i+1}(0) & \text{do not take the space} \end{cases}$$

OPF: $P_i(x_i)$ = 'take the space' if $|i| < p_{i+1} f_{i+1}(1) + (1 - p_{i+1}) f_{i+1}(0)$ otherwise 'go to the next space'.

Boundary conditions: This is given by $f_T(0) = M$, $f_T(1) = T$.

Answer: $f_{-T}(0)$ or $f_{-T}(1)$ which ever is smaller.

We now compute a numerical example. Suppose $T = 4$ and $p_i = 1 - \frac{1}{2|i|+1}$ for $i \neq 0$ and $p_0 = \frac{1}{10}$, the penalty is $M = 100$. This means that

i	-4	-3	-2	-1	0	1	2	3	4
p_i	$\frac{8}{9}$	$\frac{6}{7}$	$\frac{4}{5}$	$\frac{2}{3}$	$\frac{1}{10}$	$\frac{2}{3}$	$\frac{4}{5}$	$\frac{6}{7}$	$\frac{8}{9}$

Moreover the boundary conditions are $f_4(0) = 100$, $f_4(1) = 4$. The computation goes as follows.

Stage 3.

States	Take	Leave	$f_3(x_3)$	Decision
0	-	$\frac{8}{9} \times 4 + \frac{1}{9} \times 100 = 14\frac{2}{3}$	$14\frac{2}{3}$	leave
1	3	$\frac{8}{9} \times 4 + \frac{1}{9} \times 100 = 14\frac{2}{3}$	3	take

Stage 2.

States	Take	Leave	$f_2(x_2)$	Decision
0	-	$\frac{6}{7} \times 3 + \frac{1}{7} \times 14\frac{2}{3} = 4\frac{2}{3}$	$4\frac{2}{3}$	leave
1	2	$\frac{6}{7} \times 3 + \frac{1}{7} \times 14\frac{2}{3} = 4\frac{2}{3}$	2	take

Stage 1.

States	Take	Leave	$f_1(x_1)$	Decision
0	-	$\frac{4}{5} \times 2 + \frac{1}{5} \times 4\frac{2}{3} = 2\frac{8}{15}$	$2\frac{8}{15}$	leave
1	1	$\frac{4}{5} \times 2 + \frac{1}{5} \times 4\frac{2}{3} = 2\frac{8}{15}$	1	take

Stage 0.

States	Take	Leave	$f_0(x_1)$	Decision
0	-	$\frac{2}{3} \times 1 + \frac{1}{3} \times 2\frac{8}{15} = 1\frac{23}{45}$	$1\frac{23}{45}$	leave
1	0	$\frac{2}{3} \times 1 + \frac{1}{3} \times 2\frac{8}{15} = 1\frac{23}{45}$	0	take

Stage -1.

States	Take	Leave	$f_{-1}(x_{-1})$	Decision
0	-	$\frac{1}{10} \times 0 + \frac{9}{10} \times 1\frac{23}{45} = 1\frac{19}{25}$	$1\frac{19}{25}$	leave
1	1	$\frac{1}{10} \times 0 + \frac{9}{10} \times 1\frac{23}{45} = 1\frac{19}{25}$	1	take

Stage -2.

States	Take	Leave	$f_{-2}(x_{-2})$	Decision
0	-	$\frac{2}{3} \times 1 + \frac{1}{3} \times 1\frac{19}{25} = 1\frac{19}{75}$	$1\frac{19}{75}$	leave
1	2	$\frac{2}{3} \times 1 + \frac{1}{3} \times 1\frac{19}{25} = 1\frac{19}{75}$	$1\frac{19}{75}$	leave

Stage -3.

States	Take	Leave	$f_{-3}(x_{-3})$	Decision
0	-	$\frac{3}{4} \times 1\frac{19}{75} + \frac{1}{4} \times 1\frac{19}{75} = 1\frac{19}{75}$	$1\frac{19}{75}$	leave
1	3	$\frac{3}{4} \times 1\frac{19}{75} + \frac{1}{4} \times 1\frac{19}{75} = 1\frac{19}{75}$	$1\frac{19}{75}$	leave

Stage -4.

States	Take	Leave	$f_{-4}(x_{-4})$	Decision
0	-	$\frac{3}{4} \times 1\frac{19}{75} + \frac{1}{4} \times 1\frac{19}{75} = 1\frac{19}{75}$	$1\frac{19}{75}$	leave
1	4	$\frac{3}{4} \times 1\frac{19}{75} + \frac{1}{4} \times 1\frac{19}{75} = 1\frac{19}{75}$	$1\frac{19}{75}$	leave

Thus the optimal strategy is to drive to space -1 and proceed to take the first available space. The expected cost is $1\frac{19}{75}$.

6 The Tennis Game

You are a good tennis player. Just like any good tennis player, you have two types of serves: a hard serve (H) and a kick serve (K). The probability that H lands in bound is p_H and the probability that K lands in bound is p_K. Since the hard serve is more difficult to control, we have $p_H < p_K$.

If a hard serve lands in bound then you have a probability w_H of winning the point; and if a kick serve lands in bound then you have the probability of you winning the point is w_K with $w_K < w_H$. The problem is how should you serve to maximize your probability of winning a point?

We set up a probabilistic DP to solve this problem. The payoff is 1 if you win a point and is 0 otherwise.

1. There are a maximum of two serves (1st and 2nd serve) per point so the natural stage variable is $i = 1, 2$ corresponding to 1st and 2nd serve respectively.

2. At each stage the decision to be made is H or K.

3. Note that we do not have a state variable here.

Now let f_i = the probability of winning the point if you play optimally at stage i.

At stage 2 the probability of winning is

$$f_2 = \max \begin{cases} p_H w_H, & \text{if you serve H,} \\ p_K w_K, & \text{if you serve K.} \end{cases}$$

We have 2 cases.

Cases 1. If $p_K w_K > p_H w_H$ then $f_2 = p_K w_K$ and you should use your kick serve. Under this case, consider at stage 1, if you serve hard then we have the following table.

Event	Probability	Payoff
1st serve in and win	$p_H w_H$	1
1st serve in and lose	$p_H(1 - w_H)$	0
1st serve out	$1 - p_H$	f_2

Then the expected payoff is

$$p_H w_H \times 1 + p_H(1 - w_H) \times 0 + (1 - p_H) \times f_2 = p_H w_H + (1 - p_H) f_2.$$

If you serve kick serve then again we have

Event	Probability	Payoff
1st serve in and win	$p_K w_K$	1
1st serve in and lose	$p_K(1 - w_K)$	0
1st serve out	$1 - p_K$	f_2

The expected payoff is $p_K w_K + (1-p_K) f_2$. In this case

$$f_1 = \max \begin{cases} p_H w_H + (1-p_H) f_2 & \text{if 1st is H,} \\ p_K w_K + (1-p_K) f_2 & \text{if 1st serve is K.} \end{cases}$$

Since we assume $p_K w_K > p_H w_H$, so $f_2 = p_K w_K$ hence

$$f_1 = \max \begin{cases} p_H w_H + (1-p_H) p_k w_K & \text{if 1st is H,} \\ p_K w_K + (1-p_K) p_K w_K & \text{if 1st serve is K.} \end{cases}$$

Thus you should serve hard in the first serve if

$$p_H w_H + (1-p_H) p_K w_K \geq p_K w_K + (1-p_K) p_K w_K$$

or simply

$$\begin{aligned} p_H w_H \geq p_K w_K &= p_K^2 w_K + p_H p_K w_K \\ &= p_K w_K (1 + p_H - p_K) \end{aligned}$$

Otherwise you should use your kick serve.

Cases 2. If $p_H w_H \geq p_k w_K$ then $f_2 = p_H w_H$. A similar analysis shows that you should serve hard on the 1st serve if

$$p_H w_H + (1-p_H) p_H w_H \geq p_K w_K + (1-p_K) p_H w_H$$

or simply

$$p_H w_H (1 + p_K - p_H) \geq p_K w_K.$$

We compute a numerical example. Suppose $p_H = 0.65$, $p_K = 0.9$, $w_H = 0.8$ and $w_K = 0.65$. This is more or lass like Andre Agassi when he is good. Now

$$p_H w_H = 0.65 \times 0.8 = 0.52, \quad p_K w_K = 0.9 \times 0.65 = 0.585$$

Thus you should use your kick serve as your second serve. More over

$$p_K w_K (1 + p_H - p_K) = 0.585 \times (1 + 0.65 - 0.9) = 0.4388 < 0.52 = p_H w_H$$

thus you should use your hard serve on the first serve.