SEEM 3470: Dynamic Optimization and Applications **2013–14 Second Term**

Handout 3: Dynamic Programming Examples

Instructor: Shiqian Ma January 20, 2014

1 The discrete budgeting problem

Assume that we have to allocate a budget of size R to a series of task $\mathcal{T}$. Let a_t be the discrete action representing the amount of money allocated to task t, and let $C_t(a_t)$ be the contribution (or reward) that we receive from this allocation. We would like to maximize our total contribution

$$\begin{aligned} \max_{a} \quad & \sum_{t\in\mathcal{T}} C_t(a_t) \\ s.t., \quad & \sum_{t\in\mathcal{T}} a_t = R \\ & a_t \geq 0, \text{for all } t \in \mathcal{T}. \end{aligned}$$

We will approach this problem by first deciding how much to allocate to task 1, then to task 2, and so on, until the last task T. Let

$V_t(R_t) =$ the value of having R_t rescources remaining to allocate to task t and later tasks.

Implicit in our definition of $V_t(R_t)$ is that we are going to solve the problem of allocating R_t over tasks $t, t+1, \ldots, T$ in an optimal way. The relationship between R_{t+1} and R_t is given by

$$R_{t+1} = R_t - a_t. \tag{1}$$

R_t is known as the **state variable**. Equation (1) is the **transition function** that relates the state at time t to the state at time $t+1$. Sometimes we need to explicitly refer to the transition function (rather than just the state at time $t+1$), in which case we use

$$R^M(R_t, a_t) = R_t - a_t. \tag{2}$$

The relation between $V_t(R_t)$ and $V_{t+1}(R_{t+1})$ is given by

$$\begin{aligned} V_t(R_t) &= \max_{0\leq a_t\leq R_t} (C_t(a_t) + V_{t+1}(R^M(R_t, a_t))) \\ &= \max_{0\leq a_t\leq R_t} (C_t(a_t) + V_{t+1}(R_t - a_t)). \end{aligned} \tag{3}$$

The simplest strategy for solving our DP in (3) is to start by using $V_{T+1}(R) = 0$ (for any value of R). Then we would have

$$V_T(R_T) = \max_{0\leq a_T\leq R_T} C_T(a_T)$$

for $0 \leq R_T \leq R$. Now we know $V_T(R_T)$ for any value of R_T that might actually happen. Next we can solve

$$V_{T-1}(R_{T-1}) = \max_{0\leq a_{T-1}\leq R_{T-1}} (C_{T-1}(a_{T-1}) + V_T(R_{T-1} - a_{T-1})).$$

2 The continuous budgeting problem

Assume that the resources we are allocating are continuous (e.g., how much money to assign to various activities), which means that R_t is continuous, as is the decision of how much to budget. We assume that the contribution from allocating x_t dollars to task t is given by

$$C_t(x_t) = \sqrt{x_t}.$$

We can solve this problem exactly using DP. We first note that if we have R_T dollars left for the last task, the value of being in this state is

$$V_T(R_T) = \max_{x_T \le R_T} \sqrt{x_T}.$$

Since the contribution increases monotonically with x_T, the optimal solution is $x_T = R_T$, which means that $V_T(R_T) = \sqrt{R_T}$. Now consider the problem at time $t = T-1$. The value of being in state R_{T-1} would be

$$V_{T-1}(R_{T-1}) = \max_{x_{T-1} \le R_{T-1}} (\sqrt{x_{T-1}} + V_T(R_T(x_{T-1}))) \tag{4}$$

where $R_T(x_{T-1}) = R_{T-1} - x_{T-1}$ is the money left over from time period $T-1$. Since we know $V_T(R_T)$, we can rewrite (4) as

$$V_{T-1}(R_{T-1}) = \max_{x_{T-1} \le R_{T-1}} (\sqrt{x_{T-1}} + \sqrt{R_{T-1} - x_{T-1}}). \tag{5}$$

We solve (5) by differentiating with respect to x_{T-1} and setting the derivative equal to zero. We get $x^*_{T-1} = \frac{1}{2}R_{T-1}$. Thus,

$$V_{T-1}(R_{T-1}) = 2\sqrt{\frac{R_{T-1}}{2}}.$$

We can continue this and we will find that a general formula is

$$V_{T-t+1}(R_{T-t+1}) = t\sqrt{\frac{R_{T-t+1}}{t}}$$

or equivalently,

$$V_t(R_t) = (T-t+1)\sqrt{\frac{R_t}{T-t+1}}.$$

The optimal value of x_t is found by solving

$$\max_{x_t}(\sqrt{x_t} + (T-t)\sqrt{\frac{R_t - x_t}{T-t}}).$$

It gives

$$x^*_t = \frac{R_t}{T-t+1}.$$

This is a very intuitive result. This shows that we want to evenly divide the available budget among all remaining tasks. This is what we would expect since all the tasks produce the same contribution.

3 A stochastic model: The gambling problem

A gambler has to determine how much of his capital he should bet on each round of a game, where he will play a total of N rounds. He will win a bet with probability p and lose with probability $q = 1-p$ (assume $q < p$). Let s^n be his total capital after n plays, $n = 1, 2, \ldots, N$, with s^0 being his initial capital. For this problem we refer to s^n as the state of the system. Let a^n be the (discrete) amount he bets in round n, where we require that $a^n \le s^{n-1}$. Our gambler wants to maximize $\log s^N$. Let

$$W^n = \begin{cases} 1 & \text{if the gambler wins the } n\text{-th game,} \\ 0 & \text{otherwise.} \end{cases}$$

The system evolves according to

$$S^n = S^{n-1} + a^n W^n - a^n(1 - W^n).$$

Let $V^n(S^n)$ be the value of having S^n dollars at the end of the n-th game. The value of being in state S^n at the end of the n-th round can be written

$$\begin{aligned} V^n(S^n) &= \max_{0 \le a^{n+1} \le S^n} \mathbb{E}\{V^{n+1}(S^{n+1}) \mid S^n\} \\ &= \max_{0 \le a^{n+1} \le S^n} \mathbb{E}\{V^{n+1}(S^n + a^{n+1}W^{n+1} - a^{n+1}(1 - W^{n+1})) \mid S^n\}. \end{aligned}$$

Here we claim that the value of being in state S^n is found by choosing the decision that maximizes the expected value of being in state S^{n+1} given what we know at the end of the n-th round. We solve this by starting at the end of the N-th trial, and assuming that we have finished with S^N dollars. The value of this is

$$V^N(S^N) = \log S^N.$$

Now step back to $n = N - 1$, where we may write

$$\begin{aligned} V^{N-1}(S^{N-1}) &= \max_{0 \le a^N \le S^{N-1}} \mathbb{E}\{V^N(S^{N-1} + a^N W^N - a^N(1 - W^N)) \mid S^{N-1}\} \\ &= \max_{0 \le a^N \le S^{N-1}} [p \log(S^{N-1} + a^N) + (1-p)\log(S^{N-1} - a^N)]. \end{aligned} \tag{6}$$

The solution is given by

$$a^N = (2p - 1)S^{N-1}.$$

Now plugging this back into (6), gives

$$\begin{aligned} V^{N-1}(S^{N-1}) &= \log S^{N-1} + p\log(2p) + (1-p)\log(2(1-p)) \\ &= \log S^{N-1} + K, \end{aligned}$$

where $K := p\log(2p) + (1-p)\log(2(1-p))$, which is a constant. It turns out that we can keep applying this same logic backward in time and obtain

$$V^n(S^n) = \log(S^n) + K^n, \text{ for all } n,$$

where K^n is some constant that can be ignored. The optimal solution is

$$a^n = (2p - 1)S^{n-1}.$$

That is, the optimal strategy at each iteration is to bet a fraction $\beta = (2p-1)$ of our current money on hand. Of course, this requires that $p > 0.5$.

4 Oven temperature control

A certain material is passed through a sequence of two ovens (see Figure 1). Denote

- x_0: initial temperature of the material
- $x_k, k = 1, 2$: temperature of the material at the exit of oven k
- $u_{k-1}, k = 1, 2$: prevailing temperature in oven k

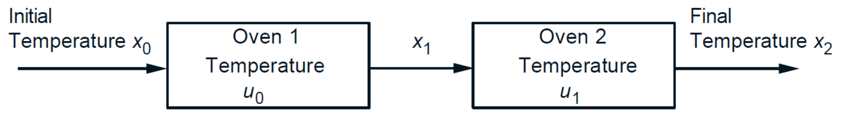

Figure 1: Example of Oven Temperature Control: The temperature of the material evolves according to $x_{k+1} = (1 - a)x_k + au_k$

We assume a model of the form

$$x_{k+1} = (1 - a)x_k + au_k, k = 0, 1,$$

where a is a known scalar from the interval $(0, 1)$. The objective is to get the final temperature x_2 close to a given target T, while expending relatively little energy. This is expressed by a cost function of the form

$$r(x_2 - T)^2 + u_0^2 + u_1^2,$$

where $r > 0$ is a given scalar. We assume no constraints on u_k. The problem is deterministic. We have $N = 2$ and a terminal cost $g_2(x_2) = r(x_2 - T)^2$, so the initial condition for the DP algorithm is

$$J_2(x_2) = r(x_2 - T)^2.$$

For the next-to-last stage, we have

$$\begin{aligned} J_1(x_1) &= \min_{u_1}[u_1^2 + J_2(x_2)] \\ &= \min_{u_1}[u_1^2 + J_2((1 - a)x_1 + au_1)]. \end{aligned}$$

Substituting the previous form of J_2, we obtain

$$J_1(x_1) = \min_{u_1}[u_1^2 + r((1 - a)x_1 + au_1 - T)^2]. \tag{7}$$

By setting the first-order derivative to be 0, we obtain

$$0 = 2u_1 + 2ra((1 - a)x_1 + au_1 - T).$$

By solving it for u_1, we obtain

$$\mu_1^*(x_1) = u_1 = \frac{ra(T-(1-a)x_1)}{1+ra^2}.$$

Note that this is not a single control but rather a control function, a rule that tells us the optimal oven temperature $u_1 = \mu_1^*(x_1)$ for each possible state x_1. By substituting the optimal u_1 in the expression (7) for J_1, we obtain

$$\begin{aligned}
J_1(x_1) &= \frac{r^2a^2((1-a)x_1-T)^2}{(1+ra^2)^2} + r\left((1-a)x_1 + \frac{ra^2(T-(1-a)x_1)}{1+ra^2} - T\right)^2 \\
&= \frac{r^2a^2((1-a)x_1-T)^2}{(1+ra^2)^2} + r\left(\frac{ra^2}{1+ra^2} - 1\right)^2 ((1-a)x_1-T)^2 \\
&= \frac{r((1-a)x_1-T)^2}{1+ra^2}.
\end{aligned}$$

We now go back one stage. We have

$$J_0(x_0) = \min_{u_0}[u_0^2 + J_1(x_1)] = \min_{u_0}[u_0^2 + J_1((1-a)x_0 + au_0)],$$

and by substituting the expression already obtained for J_1, we have

$$J_0(x_0) = \min_{u_0}\left[u_0^2 + \frac{r\left((1-a)^2x_0 + (1-a)au_0 - T\right)^2}{1+ra^2}\right].$$

By setting the first-order derivative to zero, we obtain

$$0 = 2u_0 + \frac{2r(1-a)a((1-a)^2x_0 + (1-a)au_0 - T)}{1+ra^2}.$$

This yields the optimal temperature of the first oven:

$$\mu_0^*(x_0) = \frac{r(1-a)a(T-(1-a)^2x_0)}{1+ra^2(1+(1-a)^2)}.$$

The optimal cost is obtained by substituting this expression in the formula for J_0. After lengthy calculation, this in the end yields the following formula:

$$J_0(x_0) = \frac{r((1-a)x_0-T)^2}{1+ra^2(1+(1-a)^2)}.$$

This completes the solution of the problem.

5 Inventory Control

Consider an inventory control problem, we assume that inventory x_k and the demand w_k are nonnegative integers, and that the excess demand $(w_k - x_k - u_k)$ is lost. As a result, the stock equation takes the form

$$x_{k+1} = \max(0, x_k + u_k - w_k).$$

We also assume that there is an upper bound of 2 units on the stock that can be stored, i.e., there is a constraint $x_k + u_k \leq 2$. The holding/storage cost for the k-th period is given by

$$(x_k + u_k - w_k)^2,$$

implying a penalty both for excess inventory and for unmet demand at the end of the k-th period. The ordering cost is 1 per unit stock ordered. Thus the cost per period is

$$g_k(x_k, u_k, w_k) = u_k + (x_k + u_k - w_k)^2.$$

The terminal cost is assumed to be 0, $g_N(x_N) = 0$. The planning horizon N is 3 periods, and the initial stock x_0 is 0. The demand w_k has the same probability distribution for all periods, given by

$$p(w_k = 0) = 0.1, \quad p(w_k = 1) = 0.7, \quad p(w_k = 2) = 0.2.$$

The system can also be represented in terms of the transition probabilities $p_{ij}(u)$ between the three possible states, for the different values of the control. The starting equation for the DP algorithm is

$$J_3(x_3) = 0,$$

since the terminal state cost is 0. The algorithm takes the form

$$J_k(x_k) = \min_{0 \leq u_k \leq 2 - x_k, u_k = 0,1,2} \mathbb{E}_{w_k} \left\{ u_k + (x_k + u_k - w_k)^2 + J_{k+1}(\max(0, x_k + u_k - w_k)) \right\},$$

where $k = 0, 1, 2$ and x_k, u_k, w_k can take the values 0,1, and 2.

Period 2: We compute $J_2(x_2)$ for each of the three possible states. We have

$$\begin{aligned} J_2(0) &= \min_{u_2 = 0,1,2} \mathbb{E}_{w_2}\{u_2 + (u_2 - w_2)^2\} \\ &= \min_{u_2 = 0,1,2} [u_2 + 0.1(u_2)^2 + 0.7(u_2 - 1)^2 + 0.2(u_2 - 2)^2]. \end{aligned}$$

We calculate the expectation of the right side for each of the three possible values of u_2:

$$\begin{aligned} u_2 = 0 :& \mathbb{E}\{\cdot\} = 0.7 \cdot 1 + 0.2 \cdot 4 = 1.5, \\ u_2 = 1 :& \mathbb{E}\{\cdot\} = 1 + 0.1 \cdot 1 + 0.2 \cdot 1 = 1.3, \\ u_2 = 2 :& \mathbb{E}\{\cdot\} = 2 + 0.1 \cdot 4 + 0.7 \cdot 1 = 3.1. \end{aligned}$$

Hence we have, by selecting the minimizing u_2,

$$J_2(0) = 1.3, \quad \mu_2^*(0) = 1.$$

For $x_2 = 1$, we have

$$\begin{aligned} J_2(1) &= \min_{u_2 = 0,1} \mathbb{E}_{w_2}\{u_2 + (1 + u_2 - w_2)^2\} \\ &= \min_{u_2 = 0,1} [u_2 + 0.1(1 + u_2)^2 + 0.7(u_2)^2 + 0.2(u_2 - 1)^2]. \end{aligned}$$

The expected value in the right side is

$$\begin{aligned} u_2 = 0 :& \mathbb{E}\{\cdot\} = 0.1 \cdot 1 + 0.2 \cdot 1 = 0.3, \\ u_2 = 1 :& \mathbb{E}\{\cdot\} = 1 + 0.1 \cdot 4 + 0.7 \cdot 1 = 2.1. \end{aligned}$$

Hence,

$$J_2(1) = 0.3, \quad \mu_2^*(1) = 0.$$

For $x_2 = 2$, the only admissible control is $u_2 = 0$, so we have

$$J_2(2) = \mathbb{E}_{w_2}\{(2 - w_2)^2\} = 0.1 \cdot 4 + 0.7 \cdot 1 = 1.1,$$

$$J_2(2) = 1.1, \quad \mu_2^*(2) = 0.$$

Period 1: Again we compute $J_1(x_1)$ for each of the three possible states $x_1 = 0, 1, 2$, using the values $J_2(0)$, $J_2(1)$, $J_2(2)$ obtained in the previous period. For $x_1 = 0$, we have

$$J_1(0) = \min_{u_1=0,1,2} \mathbb{E}_{w_1}\{u_1 + (u_1 - w_1)^2 + J_2(\max(0, u_1 - w_1))\},$$

$$\begin{aligned}
u_1 = 0 :& \mathbb{E}\{\cdot\} = 0.1 \cdot J_2(0) + 0.7(1 + J_2(0)) + 0.2(4 + J_2(0)) = 2.8,\\
u_1 = 1 :& \mathbb{E}\{\cdot\} = 1 + 0.1(1 + J_2(1)) + 0.7 \cdot J_2(0) + 0.2(1 + J_2(0)) = 2.5,\\
u_1 = 2 :& \mathbb{E}\{\cdot\} = 2 + 0.1(4 + J_2(2)) + 0.7(1 + J_2(1)) + 0.2 \cdot J_2(0) = 3.68,
\end{aligned}$$

$$J_1(0) = 2.5, \quad \mu_1^*(0) = 1.$$

For $x_1 = 1$, we have

$$J_1(1) = \min_{u_1=0,1} \mathbb{E}_{w_1}\{u_1 + (1 + u_1 - w_1)^2 + J_2(\max(0, 1 + u_1 - w_1))\},$$

$$\begin{aligned}
u_1 = 0 :& \mathbb{E}\{\cdot\} = 0.1(1 + J_2(1)) + 0.7 \cdot J_2(0) + 0.2(1 + J_2(0)) = 1.5,\\
u_1 = 1 :& \mathbb{E}\{\cdot\} = 1 + 0.1(4 + J_2(2)) + 0.7(1 + J_2(1)) + 0.2 \cdot J_2(0) = 2.68,
\end{aligned}$$

$$J_1(1) = 1.5, \quad \mu_1^*(1) = 0.$$

For $x_1 = 2$, the only admissible control is $u_1 = 0$, so we have

$$\begin{aligned}
J_1(2) &= \mathbb{E}_{w_1}\{(2 - w_1)^2 + J_2(\max(0, 2 - w_1))\}\\
&= 0.1(4 + J_2(2)) + 0.7(1 + J_2(1)) + 0.2 \cdot J_2(0)\\
&= 1.68,
\end{aligned}$$

$$J_1(2) = 1.68, \quad \mu_1^*(2) = 0.$$

Period 0: Here we need to compute only $J_0(0)$ since the initial state is known to be 0. We have

$$J_0(0) = \min_{u_0=0,1,2} \mathbb{E}_{w_0}\{u_0 + (u_0 - w_0)^2 + J_1(\max(0, u_0 - w_0))\},$$

$$\begin{aligned}
u_0 = 0 :& \mathbb{E}\{\cdot\} = 0.1 \cdot J_1(0) + 0.7(1 + J_1(0)) + 0.2(4 + J_1(0)) = 4.0,\\
u_0 = 1 :& \mathbb{E}\{\cdot\} = 1 + 0.1(1 + J_1(1)) + 0.7 \cdot J_1(0) + 0.2(1 + J_1(0)) = 3.7,\\
u_0 = 2 :& \mathbb{E}\{\cdot\} = 2 + 0.1(4 + J_1(2)) + 0.7(1 + J_1(1)) + 0.2 \cdot J_1(0) = 4.818,
\end{aligned}$$

$$J_0(0) = 3.7, \quad \mu_0^*(0) = 1.$$

Thus the optimal ordering policy for each period is to order one unit if the current stock is zero and order nothing otherwise.

6 Optimizing a 2-Game Chess Match Strategy and Value of Information

A player is about to play a chess match with an opponent, and wants to maximize his winning chances. The player has two playing styles to choose.

- (1) *Timid play* with which he draws with probability $p_d > 0$, and he loses with probability $(1 - p_d)$.
- (2) *Bold play* with which he wins with probability p_w, and he loses with probability $(1 - p_w)$.

Note that if the score is tied at the end of the second games, the match goes into sudden-death mode, whereby the players continue to play until the first time one of them wins a game (and the match).

Suppose the player chooses a policy of playing timid if and only if he is ahead in the score (we will see later that this policy is optimal, assuming $p_d > p_w$). Then after the first game (in which he plays bold), the score is 1-0 with probability p_w and 0-1 with probability $1 - p_w$. In the second game, he plays timid in the former case and bold in the latter case. Thus after two games, the probability of a match win is $p_w p_d$, the probability of a match loss is $(1 - p_w)^2$, and the probability of a tied score is $p_w(1 - p_d) + (1 - p_w)p_w$, in which case he has a probability p_w of winning the subsequent sudden-death game. Thus the probability of winning the match with the given strategy is

$$p_w p_d + p_w \left(p_w(1 - p_d) + (1 - p_w)p_w \right),$$

which, with some rearrangement, gives

$$\text{Probability of a match win } = p_w^2(2 - p_w) + p_w(1 - p_w)p_d. \tag{8}$$

From (8) we can see that the chance of a match win can be greater than 50-50, provided that p_w is close enough to 1/2 and p_d is close enough to 1. As an example, for $p_w = 0.45$ and $p_d = 0.9$, Eq. (8) gives a match win probability of roughly 0.53.

- **Open-Loop Minimization:** Select all controls $u_0, \ldots, u_{N-1}$ at once at time 0.
- **Closed-Loop Minimization:** Select a policy $\{\mu_0, \ldots, \mu_{N-1}\}$ that applies the control $\mu_k(x_k)$ at time k with knowledge of the current state x_k.

To calculate the value of information, let us consider the four open-loop policies, whereby we decide on the type of play to be used without waiting to see the result of the first game. These are:

- (1). Play timid in both games; this has a probability $p_d^2 p_w$ of winning the match.
- (2). Play bold in both games; this has a probability $p_w^2 + 2p_w^2(1 - p_w) = p_w^2(3 - 2p_w)$ of winning the match.
- (3). Play bold in the first game and timid in the second game; this has a probability $p_w p_d + p_w^2(1 - p_d)$ of winning the match.
- (4). Play timid in the first game and bold in the second game; this also has a probability $p_w p_d + p_w^2(1 - p_d)$ of winning the match.

This first policy is always dominated by the others, and the optimal open-loop probability of winning the match is

$$\begin{aligned}&\text{Open-loop probability of win}\\ =&\max\big(p_w^2(3-2p_w), p_w p_d + p_w^2(1-p_d)\big)\\ =&p_w^2 + p_w(1-p_w)\max(2p_w, p_d).\end{aligned} \tag{9}$$

Thus if $p_d > 2p_w$, we see that the optimal open-loop policy is to play timid in one of the two games and play bold in the other, and otherwise it is optimal to play bold in both games. For $p_w = 0.45$ and $p_d = 0.9$, (9) gives an optimal open-loop match win probability of roughly 0.425. Thus, the value of the information (the outcome of the first game) is the difference of the optimal closed-loop and open-loop values, which is approximately $0.53 - 0.425 = 0.105$. More generally, by subtracting Eq. (8) and (9), we see that

$$\begin{aligned}&\text{Value of information}\\ =&p_w^2(2-p_w) + p_w(1-p_w)p_d - p_w^2 - p_w(1-p_w)\max(2p_w, p_d)\\ =&p_w(1-p_w)\min(p_w, p_d - p_w).\end{aligned}$$

7 Optimizing an N-Game Chess Match Strategy

A player is about to play a chess match with an opponent, and wants to maximize his winning chances. The player has two playing styles to choose.

- (1) *Timid play* with which he draws with probability $p_d > 0$, and he loses with probability $(1 - p_d)$.
- (2) *Bold play* with which he wins with probability p_w, and he loses with probability $(1 - p_w)$.

Let us consider the general case of an N-game match, and let the state be the *net score*, that is, the difference between the points of the player minus the points of the opponent (so a state of 0 corresponds to an even score). Note that if the score is tied at the end of the N games, the match goes into sudden-death mode, whereby the players continue to play until the first time one of them wins a game (and the match). The optimal cost-to-go function at the start of the k-th game is given by the dynamic programming recursion

$$\begin{aligned}J_k(x_k) = \max[&p_d J_{k+1}(x_k) + (1-p_d)J_{k+1}(x_k - 1),\\ &p_w J_{k+1}(x_k + 1) + (1-p_w)J_{k+1}(x_k - 1)].\end{aligned} \tag{10}$$

The maximum above is taken over the two possible decisions:

- (a). Timid play, which keeps the score at x_k with probability p_d, and changes x_k to $x_k - 1$ with probability $1 - p_d$.
- (b). Bold play, which changes x_k to $x_k + 1$ or to $x_k - 1$ with probabilities p_w or $(1 - p_w)$, respectively.

It is optimal to play bold when

$$p_w J_{k+1}(x_k + 1) + (1-p_w)J_{k+1}(x_k - 1) \geq p_d J_{k+1}(x_k) + (1-p_d)J_{k+1}(x_k - 1)$$

or equivalently, if

$$\frac{p_w}{p_d} \geq \frac{J_{k+1}(x_k) - J_{k+1}(x_k - 1)}{J_{k+1}(x_k + 1) - J_{k+1}(x_k - 1)}. \tag{11}$$

The DP recursion is started with

$$J_N(x_N) = \begin{cases} 1 \text{ if } x_N > 0, \\ p_w \text{ if } x_N = 0, \\ 0 \text{ if } x_N < 0. \end{cases} \tag{12}$$

In this equation, we have $J_N(0) = p_w$ because when the score is even after N games ($x_N = 0$), it is optimal to play bold in the first game of sudden death.

By executing the DP algorithm (10) starting with the terminal condition (12), and using the criterion (11) for optimality of bold play, we find the following, assuming that $p_d > p_w$:

$$\begin{aligned} J_{N-1}(x_{N-1}) &= 1 \text{ for } x_{N-1} > 1; \quad \text{optimal play: either} \\ J_{N-1}(1) &= \max\,[p_d + (1-p_d)p_w, p_w + (1-p_w)p_w] \\ &= p_d + (1-p_d)p_w; \quad \text{optimal play: timid} \\ J_{N-1}(0) &= p_w; \quad \text{optimal play: bold} \\ J_{N-1}(-1) &= p_w^2; \quad \text{optimal play: bold} \\ J_{N-1}(x_{N-1}) &= 0 \text{ for } x_{N-1} < -1; \quad \text{optimal play: either.} \end{aligned}$$

Also, given $J_{N-1}(x_{N-1})$, and Eqs. (10) and (11) we obtain

$$\begin{aligned} J_{N-2}(0) &= \max\,[p_d p_w + (1-p_d)p_w^2, p_w(p_d + (1-p_d)p_w) + (1-p_w)p_w^2] \\ &= p_w(p_w + (p_w + p_d)(1-p_w)) \end{aligned}$$

and that if the score is even with 2 games remaining, it is optimal to play bold. Thus for a 2-game match, the optimal policy for both periods is to play timid if and only if the player is ahead in the score. The region of pairs (p_w, p_d) for which the player has a better than 50-50 chance to win a 2-game match is

$$R_2 = \{(p_w, p_d) \mid J_0(0) = p_w(p_w + (p_w + p_d)(1-p_w)) > 1/2\}.$$

Note that it includes points where $p_w < 1/2$.