Dynamic optimization (without risk)

Let's call the decision environment "state", which encompasses all relevant factors we may consider when making a decision (e.g. prices and budget in a standard utility maximization problem). A key feature of dynamic optimization is that

today's decision affects tomorrow's state.

Let s_t denote the state at period t. For example, a 2-period consumption smoothing problem:

$$\max_{x_1,x_2} \sum_{t=1}^{2} u(x_t) \quad \text{subject to} \quad x_1 \in (0, s_1],\ x_2 \in (0, s_2]$$

where

$$\begin{aligned} \text{Instantaneous utility function}: &\quad u(x_t) = \log x_t \\ \text{Endowment}: &\quad 100 \\ \text{Interest rate}: &\quad r \geq 0 \end{aligned}$$

So, $s_1 = 100$ and $s_2 = (1+r)(100 - x_1)$. Note that s_2 depends on x_1.

A solution, $x_1^* = 50$ and $x_2^* = 50(1+r)$, can be found by solving a familiar problem (equivalent to Cobb-Douglas utility $x_1^{0.5} x_2^{0.5}$):

$$\max_{x_1,x_2} \sum_{t=1}^{2} \log x_t \quad \text{subject to} \quad (1+r)x_1 + x_2 = (1+r)100, \tag{1}$$

we may interpret $1 + r$ as a (relative) price and $(1 + r)100$ as a budget.

But, why can we find the solution of the original dynamic problem by solving (1)? First, notice that we may rewrite the above constraint

$$\begin{aligned} &(1+r)x_1 + x_2 = (1+r)100 \\ \Leftrightarrow\ &x_2 = (1+r)(100 - x_1) \\ \Leftrightarrow\ &x_2 = s_2 \end{aligned}$$

It means that we consume everything left after choosing x_1. This is intuitive because of the monotonicity of $u(x_2) = \log x_2$. In essence, conditional on x_1, we first solve

$$x_2^* = s_2 = \operatorname*{argmax}_{x_2} \log x_2 \quad \text{subject to} \quad x_2 \in (0, s_2],$$

and then, solve

$$x_1^* = \operatorname*{argmax}_{x_1,x_2} \sum_{t=1}^{2} \log x_t \quad \text{subject to} \quad x_1 \in (0, s_1],\ x_2 = x_2^*.$$

This procedure is called backward induction.

In general, backward induction is an approach to solving a finite-horizon dynamic optimization problem. Let T be the last period. Then, backward induction finds a solution by repeatedly solving, from $t = T$ backward, a static subproblem at t conditional on the decision at $t-1$.

The following is a more general example from Sundaram (1996), which is a T-period consumption smoothing problem.

$$\max_{x_1,\ldots,x_T} \sum_{t=1}^{T} u(x_t) \quad \text{subject to} \quad x_t \in [0, s_t], \forall t \tag{2}$$

where

$$\begin{aligned} \text{Instantaneous utility function}: &\quad u(x_t) = \sqrt{x_t} \\ \text{Endowment}: &\quad s_1 \geq 0 \\ \text{Interest rate}: &\quad r \geq 0 \end{aligned}$$

So, the state at t is $s_t = (1+r)(s_{t-1} - x_{t-1})$. As above, at the last period $t = T$, we should consume everything left, i.e.

$$x_T^* = s_T = (1+r)(s_{T-1} - x_{T-1}) \text{ and get } u(x_T^*) = \sqrt{(1+r)(s_{T-1} - x_{T-1})}.$$

How about a decision at period $t = T-1$? We solve

$$\begin{aligned} &\max_{x_{T-1}\in[0,s_{T-1}]} u(x_{T-1}) + u(x_T^*) \\ \Leftrightarrow \quad &\max_{x_{T-1}\in[0,s_{T-1}]} \sqrt{x_{T-1}} + \sqrt{(1+r)(s_{T-1} - x_{T-1})}, \end{aligned}$$

which is a concave function. Hence, solving the first-order condition, we find

$$x_{T-1}^* = \frac{s_{T-1}}{1+(1+r)} \text{ and get } u(x_{T-1}^*) = \sqrt{\frac{s_{T-1}}{1+(1+r)}}.$$

Then, for $t = T-2$, we solve

$$\max_{x_{T-2}\in[0,s_{T-2}]} u(x_{T-2}) + [u(x_{T-1}^*) + u(x_T^*)].$$

You probably notice that the process can repeat until $t = 1$. Indeed, in general,

$$x_t^* = \frac{s_t}{1+(1+r)+\cdots+(1+r)^{T-t}},$$

which yields the solution to (2).

In summary, for $T < \infty$, we can solve a dynamic optimization problem of this kind by backward induction, which turns it into T number of static optimization subproblems and solves each from $t = T$ backward. At each subproblem, s_t contains all relevant factors, especially, the preceding decision x_{t-1} as a conditioning factor. Each subproblem at period t under state s_t is

$$\begin{aligned} & \max_{x_t \in [0, s_t]} u(x_t) + V_{t+1}(s_{t+1}) \\ \Leftrightarrow \quad & \max_{x_t \in [0, s_t]} u(x_t) + V_{t+1}((1+r)(s_t - x_t)), \end{aligned}$$

where $V_t(s)$ is called the value function, which takes a state and returns

$$V_t(s) = \max_{x_t \in [0, s]} u(x_t) + V_{t+1}((1+r)(s - x_t)).$$

Back to the above example, we have

$$\begin{aligned} V_T(s_T) &= \max_{x_T \in [0, s_T]} u(x_T) + 0 \\ &= \sqrt{s_T} \\ &= \sqrt{(1+r)(s_{T-1} - x_{T-1})} \\ V_{T-1}(s_{T-1}) &= \max_{x_{T-1} \in [0, s_{T-1}]} u(x_{T-1}) + V_T(s_T) \\ &= \max_{x_{T-1} \in [0, s_{T-1}]} u(x_{T-1}) + \sqrt{s_T} \\ &= \max_{x_{T-1} \in [0, s_{T-1}]} \sqrt{x_{T-1}} + \sqrt{(1+r)(s_{T-1} - x_{T-1})} \\ V_{T-2}(s_{T-2}) &= \max_{x_{T-2} \in [0, s_{T-2}]} u(x_{T-2}) + V_{T-1}(s_{T-1}) \\ &= \max_{x_{T-2} \in [0, s_{T-2}]} u(x_{T-2}) + \sqrt{(1 + (1+r))s_{T-1}} \\ &= \max_{x_{T-2} \in [0, s_{T-2}]} \sqrt{x_{T-2}} + \sqrt{(1 + (1+r))((1+r)(s_{T-2} - x_{T-2}))} \\ &\vdots \\ V_t(s) &= \sqrt{(1 + (1+r) + \cdots + (1+r)^{T-t})s} \quad \text{(in general)} \end{aligned}$$