Optimization

Lecture 14: Dynamic programming and Bellman's principle

Parin Chaipunya
KMUTT
└ Mathematics @ Faculty of Science
└ The Joint Graduate School of Energy and Environments

Areas of research:
- Multi-agent optimization: Bilevel programs, Game theory
- Optimization modeling: mainly focused on energy and environmental applications

parin.cha@kmutt.ac.th
parinchaipunya.com
github.com/parinchaipunya

Last update: Januaray 2026

Table of contents

Dynamic programming

Bellman's principle of optimality

Examples
- Pathfinding
- Saving problem

Section 1

Dynamic programming

Dynamic programming

Dynamic programming is a class of optimization problem that involves a sequence decisions at different **stages** and an evolving **states** of the system. At each stage, a decision or action is chosen based on the current state which them determines the **instantaneous cost** (or reward) and a transition into a new state.

Since future possibilities depend on the evolving states generated by earlier decisions, dynamic programming systematically evaluates these interrelated choices to identify an optimal policy over the entire planning horizon.

Stages and states

Consider a finite set of states, namely $\mathcal{T} = \{0, 1, 2, \ldots, T\}$.
Let $\mathcal{Y}_t$ denote the set of all possible states at the stage $t \in \mathcal{T}$ and $\mathcal{X}_t$ the set of all possible actions at $t \in \mathcal{T} \setminus \{T\}$.
At $t = 0, 1, \ldots, T-1$, any given $y_t \in \mathcal{Y}_t$ and action $x_t \in \mathcal{X}_t$ determines the next state $y_{t+1} \in \mathcal{Y}_{t+1}$ by the following **state-transition equation** (or **dynamical equation**)

$$y_{t+1} = f_t(y_t, x_t)\,, \tag{1}$$

where $f_t : \mathcal{Y}_t \times \mathcal{X}_t \to \mathcal{Y}_{t+1}$ is called a **state-transition mapping**.

It is important to note that the dynamical equation connects variables from different stages together.

Instantaneous cost, terminal cost, and total cost

At each stage $t = 0, 1, \ldots, T-1$, the current state y_t and an action x_t also determines an **instantaneous cost**

$$L_t(y_t, x_t)\,,$$

with $L_t : \mathcal{Y}_t \times \mathcal{X}_t \to \mathbb{R}$.

At the terminal stage T, a final state y_T is reached and a **terminal cost**

$$K_T(y_T)\,, \quad K_T : \mathcal{Y}_T \to \mathbb{R}\,,$$

is inflicted.

Hence, the total cost in a dynamic program is usually computed additive im time, which gives

$$J(y, x) = \sum_{t=0}^{T-1} L_t(y_t, x_t) + K_T(y_T)\,.$$

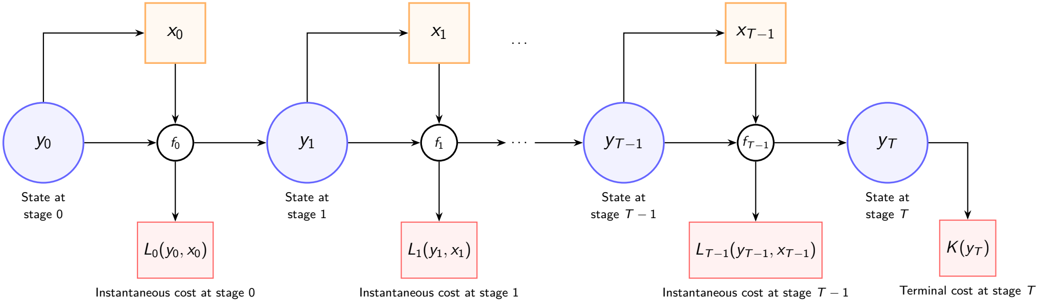
Dynamic programs

A **dynamic program**, with a known initial state $\hat{y}_0 \in \mathcal{Y}_0$, could be formulated as the following optimization problem.

$$
\begin{cases}
\min & \sum_{t=0}^{T-1} L_t(y_t, x_t) + K_T(y_T) & \text{(2a)} \\
\text{s.t.} & y_{t+1} = f_t(y_t, x_t) \quad \forall t = 0, 1, \dots, T-1 & \text{(2b)} \\
& y_0 = \hat{y}_0 & \text{(2c)} \\
& (y, x) \in \mathcal{A}\,, & \text{(2d)}
\end{cases}
$$

where $\mathcal{A} \subset \prod_{t=0}^{T} \mathcal{Y}_t \times \prod_{t=0}^{T-1} \mathcal{X}_t$ denotes the additional constraints.

Dynamic programs



Dynamic programs

Of course, this problem (2) can be treated as a classical optimization problem.

However, when the time horizon T is large, the stage-coupling system becomes more expensive to solve especially when the state-transition mappings are nonlinear, or the controls are discrete (*e.g.* integer), or some uncertainty is involved.

This motivates us to adopt a decomposition using the Bellman's principle (introduced in the next section), which is not only cheaper, but also exploit the sequential nature of decision making. Afterall, the Bellman's principle eventually brings a better insight to the dynamic programming problems.

An example

- At the start of the year, a man decides how much money to use each month.
- The remaining wealth is saved and earns interests.
- The consumption brings satisfaction and so does the wealth at the end of the year.
- What would be his annual financial plan if he wishes to maximize his satisfaction?

An example

Let

- y_t denote the wealth at time $t \in \{0, 1, \ldots, 12\}$ (that is, $T = 12$),
- x_t denote the consumption at time $t \in \{0, 1, \ldots, 11\}$,
- w_t denote the external income (*e.g.* salary),
- r denote the interest rate,
- $L_t(y_t, x_t) = U(x_t)$ denote the satisfaction of consuming x_t,
- $K_T(y_T) = S(y_T)$ denote the remaining savings at year end.

An example

A dynamic programming formulation is

$$
\begin{cases}
\max & \displaystyle\sum_{t=0}^{11} U(x_t) + S(y_{12}) \\
\text{s.t.} & y_{t+1} = (1+r)(y_t - x_t) + w_t \qquad \forall t = 0, 1, \ldots, 11 \\
& 0 \leq x_t \leq y_t \qquad \forall t = 0, 1, \ldots, 11 \\
& y_0 = \hat{y}_0 \, .
\end{cases}
$$

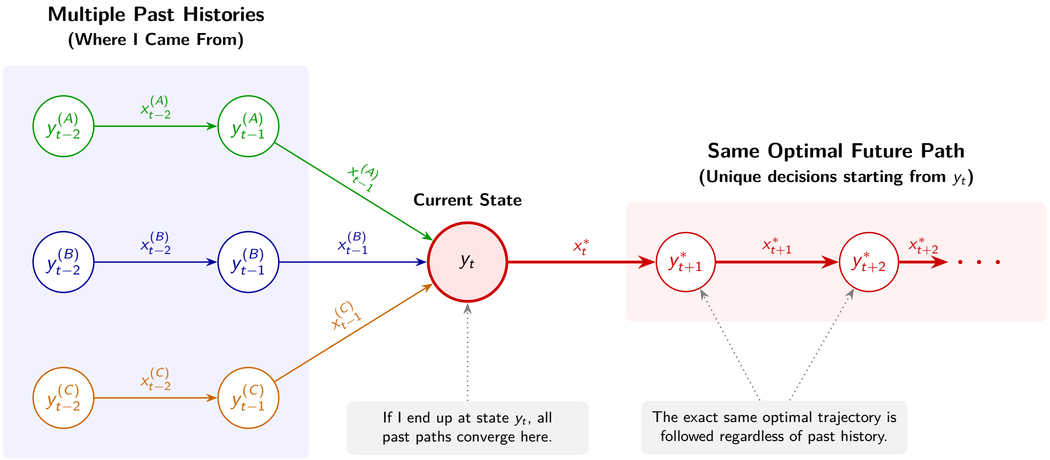
Section 2

Bellman's principle of optimality

Bellman's principle of optimality

Bellman's principle of optimality.
An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.

Bellman's principle of optimality



Bellman's principle of optimality

In the Bellman's approach, we do not only solve the dynamic program (2) with a fixed initial condition. Rather, we shall look for an **optimal policy** $\pi_t : \mathcal{Y}_t \to \mathcal{X}_t$, that provides an optimal action from any given state at any given stage.

Bellman functions and the optimal policies

We define the **Bellman value function** (or simply **Bellman function**) at each stage $t \in \{0, 1, \dots, T\}$ in a backward induction as follows.

$$\begin{aligned} V_T(y_T) &= K_T(y_T)\,, \\ V_t(y_t) &= \inf_{x_t \in \Gamma_t(y_t)} [L_t(y_t, x_t) + V_{t+1}(f(y_t, x_t))]\,, \qquad \forall t = 0, 1, \dots, T-1\,, \end{aligned}$$

where $\Gamma_t(y_t)$ denotes the **state-induced constraint** at the stage t.

At any $t = 0, 1, \dots, T-1$, we define the **optimal policy** $\pi_t : \mathcal{Y}_t \to \mathcal{X}_t$ from the condition

$$\pi_t(y_t) \in \operatorname*{arg\,min}_{x_t \in \Gamma_t(y_t)} [L_t(y_t, x_t) + V_{t+1}(f(y_t, x_t))]\,.$$

That is, the optimal policy $\pi_t(y_t)$ is a minimizer that produces the value V_t.

A closer look

Starting from the terminal stage T, the cost is straightforward and hence the Bellman value function is simply the terminal cost $V_T = K_T$.

Stepping back to the stage $T-1$ at any state y_{T-1}, any action x_{T-1} inflcits an instataneous cost $L_{T-1}(y_{T-1}, x_{T-1})$ and the state is transitioned into the final state $f_{T-1}(y_{T-1}, x_{T-1})$, which additionally costs $K_T(f_{T-1}(y_{T-1}, x_{T-1}))$. The Bellman function V_{T-1} simply measures the least forward cost possible starting from y_{T-1} at $T-1$.

In fact, for any $t = 0, 1, \ldots, T-1$ and a current state y_t, an action x_t generates an instataneous cost $L_t(y_t, x_t)$ and a successive state $f_t(y_t, x_t)$. The least cost from this future state forward is given by $V_{t+1}(f_t(y_t, x_t))$. Since the cost is additive, the least cost from the current state y_t is given by the Bellman function

$$V_t(y_t) = \overbrace{\inf_{x_t \in \Gamma_t(y_t)} \Big[\underbrace{L_t(y_t, x_t)}_{\text{Current cost}} + \underbrace{V_{t+1}(f(y_t, x_t))}_{\text{Least future cost}} \Big]}^{\text{Least cost starting at } y_t} .$$

Section 3

Examples

Subsection 1

Pathfinding

Pathfinding

We want to move from an initial point A to a terminal point F passing through the following layers

$$A \to \{B, C\} \to \{D, E\} \to F \,.$$

The distances from one node to another is presented in the following table.

Table: The distance of traveling between feasible nodes.

	A	B	C	D	E	F
A		2	4			
B				3	6	
C				1	2	
D						5
E						1

This is a dynamic programming problem with $T = 3$ and decision makings at $t = 0, 1, 2$.

Pathfinding

Decisions

$\mathcal{X}_t = \{A, B, C, D, E, F\}$	The set of all nodes at stage t.
$\Gamma_0 = \{B, C\}$	Feasible nodes at stage $y = 0$.
$\Gamma_1 = \{D, E\}$	Feasible nodes at stage $y = 1$.
$\Gamma_2 = \{F\}$	Feasible nodes at stage $y = 2$.
$x_t \in \Gamma_t$	The node to visit at stage t.

Pathfinding

States	
$\mathcal{Y}_0 = \{A\}$	The possible states at $y = 0$.
$\mathcal{Y}_1 = \{B, C\}$	The possible states at $y = 1$.
$\mathcal{Y}_2 = \{D, E\}$	The possible states at $y = 2$.
$\mathcal{Y}_3 = \{F\}$	The possible states at $y = 3$.

The state-transition equation for each stage t is given as

$$f_t(y_t, x_t) = x_t \, .$$

Pathfinding

Costs
The instantaneous cost $L_t(y_t, x_t)$ is given by the distance table. In particular,

$$\begin{aligned} L_0(A,B) &= 2 & L_0(A,C) &= 4 \\ L_1(B,D) &= 3 & L_1(B,E) &= 6 \\ L_1(C,D) &= 1 & L_1(C,E) &= 2 \\ L_2(D,F) &= 5 & L_2(E,F) &= 1\,. \end{aligned}$$

There is no terminal cost here, hence $K_3 = 0$.

Pathfinding

Now that all the ingredients are ready, we start the analysis following the Bellman's principle.

At the terminal stage $t = T = 3$, the only possible state here is $y_3 = F$. Thus, we have

$$V_3(F) = K_3(F) = 0\,.$$

Pathfinder

Next we condiser the stage $t = 2$, where we have $\mathcal{Y}_2 = \{D, E\}$ and $\Gamma_2 = \{F\}$.

At the state $y_2 = D$, we get

$$\begin{aligned} V_2(D) &= \inf_{x_2 \in \Gamma_2} [L_2(D, x_2) + V_3(f_2(D, x_2))] \\ &= L_2(D, F) + V_3(F) = 5 + 0 = 5\,. \end{aligned}$$

We also obtain $\pi_2(D) = F$.

Similarly, we have

$$V_2(E) = L_2(E, F) + V_3(F) = 1 + 0 = 1\,,$$

and $\pi_2(E) = F$.

This yields an optimal action table.

	$y_2 = D$	$y_2 = F$
$t = 2$	F	F

Pathfinder

Consider $t = 1$. We have $Y_1 = \{B, C\}$ and $\Gamma_2 = \{D, E\}$.

At the state $y_1 = B$, we get

$$\begin{aligned} V_1(B) &= \inf_{x_1 \in \Gamma_1} [L_1(B, x_1) + V_2(f_1(B, x_1))] \\ &= \min \Big\{ L_1(B, D) + V_2(D) \ , \ L_1(B, E) + V_2(E), \Big\} = \min\{3 + 5 \ , \ 6 + 1\} = 7 \, , \end{aligned}$$

with $\pi_1(B) = E$.

In the same way, we get

$$\begin{aligned} V_1(C) &= \inf_{x_1 \in \Gamma_1} [L_1(C, x_1) + V_2(f_1(C, x_1))] \\ &= \min \Big\{ L_1(C, D) + V_2(D) \ , \ L_1(C, E) + V_2(E), \Big\} = \min\{1 + 5 \ , \ 2 + 1\} = 3 \, , \end{aligned}$$

with $\pi_1(C) = E$.

Pathfinder

Coming back to the initial stage $t = 0$, we have $Y_0 = \{A\}$ and $\Gamma_0 = \{B, C\}$.

We get

$$V_0(A) = \inf_{x_0 \in \Gamma_0} [L_0(A, x_0) + V_1(f_0(A, x_0))]$$
$$= \min\left\{L_0(A, B) + V_1(B)\ ,\ L_0(A, C) + V_1(C),\right\} = \min\{2 + 7\ ,\ 4 + 3\} = 7\,,$$

which means $\pi_0(A) = C$.

Pathfinder

The above analysis constitutes the optimal policies π_t's which could be summarized as the following table of optimal actions.

state

stage		A	B	C	D	E	F
	0	C					
	1		E	E			
	2				F	F	

This table tells us that the optimal route from A to F is

$$A \to C \to E \to F\,.$$

Subsection 2

Saving problem

Saving problem

Let us consider the saving problem presented in the first section of the slides, but only with $T = 2$. Moreover, let us consider the following utility functions

$$L_t(y_t, x_t) = U(x_t) = \log(x_t)\,, \qquad \forall t = 0, 1\,,$$
$$K_T(y_T) = S(y_T) = \frac{1}{2}\log(y_T)\,.$$

We also have the constraints

$$\Gamma_t(y_t) = [0, y_t]\,, \qquad \forall t = 0, 1\,.$$

We assume that there is no income, that is

$$w_0 = w_1 = 0\,.$$

The state-transition equation then reduces to

$$y_{t+1} = (1 + r)(y_t - x_t)\,.$$

Saving problem

At $t = T = 2$, we have

$$V_2(y_2) = \frac{1}{2}\log(y_2)\,.$$

Saving problem

Next, consider the stage $t = 1$. We have

$$V_1(y_1) = \sup_{0 \le x_1 \le y_1} \Big[\underbrace{\log(x_1) + \frac{1}{2} \log\Big((1+r)(y_1 - x_1)\Big)}_{=: \, \phi_1(x_1)} \Big].$$

To find the supremum (in fact, a maximum), we find a maximizer of ϕ_1 over $\Gamma_1(y_1) = [0, y_1]$. It turns out that

$$\phi_1'(x_1^*) = 0 \iff x_1^* = \frac{y_1}{2} \quad \text{and} \quad \phi_1'(x_1) > 0 \iff 0 \le x_1 < x_1^* \, .$$

Since $x_1^* \in \Gamma_1(y_1)$, the maximizer of ϕ_1 over $\Gamma_1(y_1)$ is

$$\pi_1(y_1) = \frac{y_1}{2} \, ,$$

which yields

$$V_1(y_1) = \phi_1(\pi_1(y_1)) = \frac{3}{2} \log\left(\frac{y_1}{2}\right) + \frac{1}{2} \log(1+r) \, .$$

Saving problem

We next consider the stage $t = 0$. We have

$$V_0(y_0) = \sup_{0 \le x_0 \le y_0} \Bigg[\underbrace{\log(x_0) + \frac{3}{2} \log\left(\frac{(1+r)(y_0 - x_0)}{2} \right) + \frac{1}{2}\log(1+r)}_{=:\, \phi_0(x_0)} \Bigg]$$

With a similar idea as before, we note that

$$\phi_0'(x_0^*) = 0 \iff x_0^* = \frac{y_0}{4} \quad \text{and} \quad \phi_1'(x_0) > 0 \iff 0 \le x_0 < x_0^* \,.$$

Since $x_0^* \in \Gamma_0(y_0)$, the maximizer of ϕ_0 over $\Gamma_0(y_0)$ is

$$\pi_0(y_0) = \frac{y_0}{4} \,,$$

which then yields

$$V_0(y_0) = \phi_0(\pi_0(y_0)) = \frac{5}{2} \log\left(\frac{y_0}{4}\right) + \frac{3}{2} \log\left(\frac{3(1+r)}{2} \right) + \frac{1}{2} \log(1+r) \,.$$

Saving problem

From this analysis, we may conclude that:

- From an initial saving y_0, the optimal consumption of the month $t = 0$ is $x_0^* = \frac{y_0}{4}$.
- This leaves $y_1^* = (1+r)\frac{3y_0}{4}$.
- Then the consumption of the month $t = 1$ is $x_1^* = \frac{y_1^*}{2}$.
- This leaves $y_2^* = (1+r)\frac{y_1^*}{2}$.
- The maximum utility is reached at

$$\sup_{\substack{(y,x)\in\mathcal{A} \\ y_{t+1}=f_t(y_t,x_t)}} \left[L_0(t_0,x_0) + L_1(t_1,x_1) + K_2(t_2)\right]$$

$$= V_0(y_0) = \frac{5}{2}\log\left(\frac{y_0}{4}\right) + \frac{3}{2}\log\left(\frac{3(1+r)}{2}\right) + \frac{1}{2}\log(1+r)\,.$$

That's it!

Key takeaways.

- The concept of Bellmans' principle of optimality.
- The backward induction calculation.

Thank you.