

An introduction to dynamic programming in discrete time

Martin Branda

Charles University
Faculty of Mathematics and Physics
Department of Probability and Mathematical Statistics

COMPUTATIONAL ASPECTS OF OPTIMIZATION

Content

- 1 Two-stage stochastic programming problems
- 2 Dynamic programming and Bellman principle
- 3 Example: Cake eating problem
- 4 Example: Gambling game

Two-stage linear programming problems

$$\begin{aligned} \min \quad & c^T x + q^T y \\ \text{s.t.} \quad & Ax = b, \\ & Tx + Wy = h, \\ & x \geq 0, \\ & y \geq 0. \end{aligned} \tag{1}$$

Two-stage linear prog. problems with random parameter

$$\begin{aligned} \min \quad & c^T x + q(\xi)^T y(\xi) \\ \text{s.t.} \quad & Ax = b, \\ & T(\xi)x + Wy(\xi) = h(\xi), \\ & x \geq 0, \\ & y \geq 0, \end{aligned} \tag{2}$$

where ξ is a random vector on probability space (Ω, \mathcal{F}, P) , i.e. q, T, h as well as y are random.

Two-stage stochastic linear programming problems

$$\begin{aligned} \min \quad & c^T x + \mathbb{E} \left[q(\xi)^T y(\xi) \right] \\ \text{s.t.} \quad & Ax = b, \\ & T(\xi)x + Wy(\xi) = h(\xi), \\ & x \geq 0, \\ & y(\xi) \geq 0, \end{aligned} \tag{3}$$

where ξ is a random vector on probability space (Ω, \mathcal{F}, P) , i.e. q , T , h as well as y are random.

Two-stage stochastic linear programming problems

$$\begin{aligned} \min \quad & c^T x + \mathbb{E}[f(x, \xi)] \\ \text{s.t.} \quad & Ax = b, \\ & x \geq 0, \end{aligned} \tag{4}$$

where the second-stage problem is

$$\begin{aligned} f(x, \xi) = \min \quad & q(\xi)^T y(\xi) \\ \text{s.t.} \quad & T(\xi)x + Wy(\xi) = h(\xi), \\ & y(\xi) \geq 0, \end{aligned} \tag{5}$$

Transportation problem – deterministic

- x_{ij} – decision variable: amount transported from i to j
- c_{ij} – costs for transported unit
- a_i – capacity
- b_j – demand

ASS. $\sum_{i=1}^n a_i \geq \sum_{j=1}^m b_j$.

$$\begin{aligned} \min \quad & \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij} \\ \text{s.t.} \quad & \sum_{j=1}^m x_{ij} \leq a_i, \quad i = 1, \dots, n, \\ & \sum_{i=1}^n x_{ij} \geq b_j, \quad j = 1, \dots, m, \\ & x_{ij} \geq 0. \end{aligned}$$

Transportation problem – stochastic

- B_j – **random demand** with known distribution, $\mathbb{E}[B_j] < \infty$
- q_j^+, q_j^- – price for unit shortage, surplus (deterministic)

Two-stage problem **with simple recourse**

$$\min_{x_{ij}, y_j^\pm} \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij} + \mathbb{E} \left[\sum_{j=1}^m q_j^+ y_j^+(\xi) + q_j^- y_j^-(\xi) \right]$$

$$\text{s.t. } \sum_{j=1}^m x_{ij} \leq a_i, \quad i = 1, \dots, n,$$

$$x_{ij} \geq 0,$$

$$y_j^+(\xi) - y_j^-(\xi) = B_j(\xi) - \sum_{i=1}^n x_{ij}, \quad j = 1, \dots, m,$$

$$y_j^+(\xi), y_j^-(\xi) \geq 0.$$

Transportation problem – stochastic

- B_j – **random demand** with known distribution, $\mathbb{E}[B_j] < \infty$
- q_j^+ , q_j^- – price for unit shortage, surplus

Two-stage problem **with simple recourse**

$$\min \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij} + \mathbb{E} \left[\sum_{j=1}^m q_j^+ \left(B_j - \sum_{i=1}^n x_{ij} \right)_+ + \sum_{j=1}^m q_j^- \left(B_j - \sum_{i=1}^n x_{ij} \right)_- \right]$$

$$\text{s.t. } \sum_{j=1}^m x_{ij} \leq a_i, \quad i = 1, \dots, n,$$

$$x_{ij} \geq 0.$$

Transportation problem – stochastic

Two-stage problem with simple recourse

$$\begin{aligned} \min \quad & \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij} + \\ & + \sum_{j=1}^m q_j^+ \mathbb{E} \left[\left(B_j - \sum_{i=1}^n x_{ij} \right)_+ \right] + \sum_{j=1}^m q_j^- \mathbb{E} \left[\left(B_j - \sum_{i=1}^n x_{ij} \right)_- \right] \\ \text{s.t.} \quad & \sum_{j=1}^m x_{ij} \leq a_i, \quad i = 1, \dots, n, \\ & x_{ij} \geq 0. \end{aligned}$$

Explicit formula for $\mathbb{E} \left[(B - x)_+ \right] \dots$

Transportation problem – stochastic

Explicit formula for $\mathbb{E} \left[(B - x)_+ \right]$ under uniform distribution $B \sim U(\underline{b}, \bar{b})$

- $= 0$, if $x \geq \bar{b}$,
- $= \mathbb{E}[B] - x$, if $x \leq \underline{b}$,
- if $x \in (\underline{b}, \bar{b})$, then

$$\begin{aligned} \mathbb{E} \left[(B - x)_+ \right] &= \frac{1}{\bar{b} - \underline{b}} \int_x^{\bar{b}} z - x \, dz \\ &= \frac{1}{\bar{b} - \underline{b}} \left(\left[\frac{z^2}{2} \right]_x^{\bar{b}} - x [z]_x^{\bar{b}} \right) \\ &= \frac{(\bar{b} - x)^2}{2(\bar{b} - \underline{b})}. \end{aligned}$$

Transportation problem – stochastic

Explicit formula for $\mathbb{E} \left[(B - x)_+ \right]$ under normal distribution $B \sim N(\mu, \sigma^2)$

$$\begin{aligned} \mathbb{E} \left[(B - x)_+ \right] &= \int_x^\infty b - x dF(b) \\ \int_x^\infty b dF(b) &= \int_{\frac{x-\mu}{\sigma}}^\infty (\mu + z\sigma) \varphi(z) dz \\ &= \int_{\frac{x-\mu}{\sigma}}^\infty \mu \varphi(z) dz + \int_{\frac{x-\mu}{\sigma}}^\infty z \frac{\sigma}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} dz \\ &= \mu \left(1 - \Phi \left(\frac{x - \mu}{\sigma} \right) \right) + \frac{\sigma}{\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \\ \int_x^\infty x dF(b) &= x \left(1 - \Phi \left(\frac{x - \mu}{\sigma} \right) \right) \end{aligned}$$

Transportation problem – stochastic

Let the distribution of B_j be **finite discrete** b_{js} , $s = 1, \dots, S$, with probabilities p_s :

$$\min_{x_{ij}, y_{js}} \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij} + \sum_{s=1}^S p_s \left[\sum_{j=1}^m q_j^+ y_{js}^+ + q_j^- y_{js}^- \right]$$

$$\text{s.t. } \sum_{j=1}^m x_{ij} \leq a_i, \quad i = 1, \dots, n,$$

$$x_{ij} \geq 0,$$

$$y_{js}^+ - y_{js}^- = b_{js} - \sum_{i=1}^n x_{ij}, \quad j = 1, \dots, m, \quad s = 1, \dots, S,$$

$$y_{js}^+, y_{js}^- \geq 0.$$

Benders decomposition (L-shaped algorithm)

Master problem

$$\begin{aligned} \min_{x_{ij}} \quad & \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij} + \sum_{s=1}^S p_s f^s(x) \\ \text{s.t.} \quad & \sum_{j=1}^m x_{ij} \leq a_i, \quad i = 1, \dots, n, \\ & x_{ij} \geq 0. \end{aligned}$$

Slave (second-stage) problems

$$\begin{aligned} f^s(x) = \min_{y_{js}} \quad & \sum_{j=1}^m q_j^+ y_{js}^+ + q_j^- y_{js}^- \\ \text{s.t.} \quad & y_{js}^+ - y_{js}^- = b_{js} - \sum_{i=1}^n x_{ij}, \quad j = 1, \dots, m, \\ & y_{js}^+, y_{js}^- \geq 0. \end{aligned}$$

Content

- 1 Two-stage stochastic programming problems
- 2 Dynamic programming and Bellman principle**
- 3 Example: Cake eating problem
- 4 Example: Gambling game

Dynamic programming – Bellman principle

Reward maximization problem:

$$\begin{aligned} \max_{c \in \mathcal{C}, s \in \mathcal{S}} \mathbb{E}_W \left[\sum_{t=1}^{T \vee \infty} v^{t-1} r(s_{t-1}, c_t) \right] \\ \text{s.t. } s_t = f(s_{t-1}, c_t, W_t), \quad t = 1, \dots, T \vee \infty. \end{aligned} \quad (6)$$

- $s_t \in \mathcal{S}$ – **state variables** at the end of period t ,
- $c_t \in \mathcal{C}$ – **control variables**,
- $r : \mathcal{S} \times \mathcal{C} \rightarrow \mathbb{R}$ – **reward** under state s_{t-1} and control c ,
- v – **discount factor**,
- $f : \mathcal{S} \times \mathcal{C} \times \mathcal{W} \rightarrow \mathcal{S}$ – **transition function**
- W_t – **random variable** = information random until period t , i.e., $W_t \in \mathcal{F}_t$ for a filtration (\mathcal{F}_t) .

Dynamic programming – Bellman principle

Bellman equation

$$\begin{aligned}
 V_t(s_{t-1}) &= \max_{c_t \in \mathcal{C}} r(s_{t-1}, c_t) + v \mathbb{E}_{W_t} [V_{t+1}(s_t) | s_{t-1}] \\
 \text{s.t. } s_t &= f(s_{t-1}, c_t, W_t)
 \end{aligned} \tag{7}$$

- $s_t \in \mathcal{S}$ – **state variables** at the end of period t ,
- $c_t \in \mathcal{C}$ – **control variables**,
- $r : \mathcal{S} \times \mathcal{C} \rightarrow \mathbb{R}$ – **reward** under state s_{t-1} and control c ,
- v – **discount factor**,
- $f : \mathcal{S} \times \mathcal{C} \times \mathcal{W} \rightarrow \mathcal{S}$ – **transition function**
- W_t – **random variable** = information random until period t , i.e., $W_t \in \mathcal{F}_t$ for a filtration $(\mathcal{F}_t)_t$.
- V_t – **value function** (cost-to-go function)

Dynamic programming

Consider $|\mathcal{C}| = 10$, and $|\mathcal{S}| = 10$. Then the reward maximization problem grows by a factor 100 every time stage. Represent the dynamics as a decision tree over 10 periods. Full enumeration leads to 10^{20} nodes.

Whereas, dynamic programming collapses all the nodes of the tree that represent the same state.

Dynamic programming – Example 1

Finite T or infinite ∞ time horizon

$$\begin{aligned} \max_{s_t, c_t} \quad & \sum_{t=1}^{T \vee \infty} v^{t-1} u(c_t) \\ \text{s.t.} \quad & \\ & s_t = (1 + r)s_{t-1} + Y_t - c_t. \end{aligned} \tag{8}$$

- u – **utility function**
- s_t – **state variables** representing total amount of resources available to the consumer (initial state s_0).
- c_t – **control variables** maximizing the consumer's utility. It affects the resources available in the next period.
- Y_t – **exogenous income** (deterministic for now)
- $v = 1/(1 + i)$ – discount factor, r – exogenous interest rate

Dynamic programming

If we assume that there is a finite terminal period T :

$$\begin{aligned}
 V_1(s_0) &= \max_{s_t, c_t} \sum_{t=1}^T v^{t-1} u(c_t) \\
 &= \max_{s_t, c_t} u(c_1) + v u(c_2) + \dots + v^{T-1} u(c_T) \\
 &= \max_{s_t, c_t} u(c_1) + v \left[\sum_{t=2}^T v^{t-2} u(c_t) \right] \\
 &\text{s.t.} \\
 &s_t = (1+r)s_{t-1} + Y_t - c_t.
 \end{aligned}$$

Dynamic programming

We rewrite the maximization problem recursively and obtain the **Bellman equation**

$$V_t(s_{t-1}) = \max_{s_t, c_t} u(c_t) + v V_{t+1}(s_t),$$

where $s_t = (1 + r)s_{t-1} + Y_t - c_t$.

Moreover, since u does not depend on the time period, we can write

$$\begin{aligned} V(s_{t-1}) &= \max_{s_t, c_t} u(c_t) + v V(s_t), \\ &= \max_{c_t} u(c_t) + v V\left((1 + r)s_{t-1} + Y_t - c_t\right), \end{aligned}$$

with $V_{T+1}(s_t) = V(s_T) \equiv 0$.

Bellman 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, i.e.,

$$\hat{c}_t(s_{t-1}) \in \arg \max_{c_t} u(c_t) + v V\left((1+r)s_{t-1} + Y_t - c_t\right).$$

This principle is applied recursively (forward/backward recursion)...

Dynamic programming

First order optimality conditions

$$\frac{\partial V(s_{t-1})}{\partial c_t} = 0, \quad \frac{\partial V(s_{t-1})}{\partial s_{t-1}} = 0.$$

In particular,

$$\begin{aligned} \frac{\partial V(s_{t-1})}{\partial c_t} &= u'(c_t) + v V'(s_t) \frac{\partial s_t}{\partial c_t}, \\ \frac{\partial V(s_{t-1})}{\partial s_{t-1}} &= v V'(s_t) \frac{\partial s_t}{\partial s_{t-1}}, \end{aligned}$$

where using $s_t = (1+r)s_{t-1} + Y_t - c_t$ we have

$$\frac{\partial s_t}{\partial c_t} = -1, \quad \frac{\partial s_t}{\partial s_{t-1}} = 1 + r.$$

A stochastic extension

Let $Y_t(\xi)$ be random. We can obtain the **Bellman equation**

$$V(s_{t-1}) = \max_{c_t} u(c_t) + v \mathbb{E}_{Y_t \bullet} \left[V\left((1+r)s_{t-1} + Y_t(\xi) - c_t\right) \right].$$

Content

- 1 Two-stage stochastic programming problems
- 2 Dynamic programming and Bellman principle
- 3 Example: Cake eating problem**
- 4 Example: Gambling game

Dynamic programming – Example 2: Cake eating problem

Cake eating problem:

- $u(c) = 2c^{1/2}$,
- $s_0 = 1, s_T = 0$,
- $s_t = s_{t-1} - c_t \dots$

Example: Cake eating problem

Bellman equation

$$V(s_{t-1}) = \max_{c_t} u(c_t) + v V(s_t),$$

$$\text{s.t. } s_t = s_{t-1} - c_t.$$

Optimality conditions

$$\frac{\partial V(s_{t-1})}{\partial c_t} = u'(c_t) + v V'(s_t) \frac{\partial s_t}{\partial c_t} = 0,$$

$$\frac{\partial V(s_{t-1})}{\partial s_{t-1}} = v V'(s_t) \frac{\partial s_t}{\partial s_{t-1}} = 0,$$

From $s_t = s_{t-1} - c_t$

$$\frac{\partial s_t}{\partial c_t} = -1, \quad \frac{\partial s_t}{\partial s_{t-1}} = 1. \tag{9}$$

Example: Cake eating problem

Putting them together, we obtain

$$\frac{\partial V(s_{t-1})}{\partial c_t} = u'(c_t) - v V'(s_t) = 0, \quad (10)$$

$$\frac{\partial V(s_{t-1})}{\partial s_{t-1}} = v V'(s_t) = 0, \quad (11)$$

Taking (10) for $t - 1$

$$u'(c_{t-1}) - v V'(s_{t-1}) = 0, \quad (12)$$

and plugging it into (11), we have

$$u'(c_{t-1}) = v u'(c_t),$$

which represents the **optimal path of the cake consumption**.

Example: Cake eating problem

For $u(c) = 2c^{1/2}$, we have

$$\begin{aligned}u'(c_{t-1}) &= v u'(c_t), \\(c_{t-1})^{-1/2} &= v (c_t)^{-1/2}, \\c_t &= v^2 c_{t-1},\end{aligned}$$

with initial and terminal conditions $s_0 = 1$, $s_T = 0$. If we denote $\beta = v^2$, we obtain

$$c_t = \beta c_{t-1} = \beta^{t-1} c_1.$$

Using

$$c_t = \beta c_{t-1} = \beta^{t-1} c_1.$$

and

$$s_0 - c_1 - c_2 - \dots - c_T = s_T = 0,$$

we have

$$(1 - \beta - \dots - \beta^{T-1})c_1 = s_0,$$

and finally **optimal consumption**

$$\hat{c}_1 = \frac{1 - \beta}{1 - \beta^T} s_0,$$

$$\hat{c}_t = \beta \hat{c}_{t-1} = \beta^{t-1} \hat{c}_1.$$

Content

- 1 Two-stage stochastic programming problems
- 2 Dynamic programming and Bellman principle
- 3 Example: Cake eating problem
- 4 Example: Gambling game

Gambling game

The gambler wins/looses a bet. The goal is to maximize the probability that the gambler reaches at least bankroll G .

- T – number of periods (games)
- b_t – **bet** at period (game) t
- s_t – state of **bankroll** at the end of period t
- p – **winning probability**
- G – **bankroll goal**

Recursion

$$V_t(s) = \max_{0 \leq b_t \leq s} p V_{t+1}(s + b_t) + (1 - p) V_{t+1}(s - b_t),$$

with boundary conditions $V_T(s) = 0$ for $s \leq \lfloor G/2 \rfloor$, $V_T(s) = 1$ for $s \geq G$, and $V_T(s) = p$ otherwise.

The value function expresses the accumulated probability of reaching the goal G from actual state s .

Gambling game

Let

- $T = 3$ – number of periods (games)
- $s_0 = 2$ – state of bankroll at the beginning
- $p = 0.4$ – probability of winning
- $G = 4$ – goal bankroll

Forward recursion

b	$V_1(2)$
0	$0.4V_2(2 + 0) + 0.6V_2(2 - 0)$
1	$0.4V_2(2 + 1) + 0.6V_2(2 - 1)$
2	$0.4V_2(2 + 2) + 0.6V_2(2 - 2)$

$$b \quad V_2(0)$$

$$0 \quad 0.4V_3(0 + 0) + 0.6V_3(0 - 0)$$

$$b \quad V_2(1)$$

$$0 \quad 0.4V_3(1 + 0) + 0.6V_3(1 - 0)$$

$$1 \quad 0.4V_3(1 + 1) + 0.6V_3(1 - 1)$$

$$b \quad V_2(2)$$

$$0 \quad 0.4V_3(2 + 0) + 0.6V_3(2 - 0)$$

$$1 \quad 0.4V_3(2 + 1) + 0.6V_3(2 - 1)$$

$$2 \quad 0.4V_3(2 + 2) + 0.6V_3(2 - 2)$$

$$b \quad V_2(3)$$

$$0 \quad 0.4V_3(3 + 0) + 0.6V_3(3 - 0)$$

$$1 \quad 0.4V_3(3 + 1) + 0.6V_3(3 - 1)$$

$$b \quad V_2(4)$$

$$0 \quad 0.4V_3(4 + 0) + 0.6V_3(4 - 0)$$

Gambling game

Boundary conditions

$$V_3(0) = 0$$

$$V_3(1) = 0$$

$$V_3(2) = 0.4$$

$$V_3(3) = 0.4$$

$$V_3(4) = 1$$

$$\begin{array}{r}
 b \\
 0
 \end{array}
 \quad
 \begin{array}{r}
 V_2(0) \\
 0.4V_3(0+0) + 0.6V_3(0-0) = 0.4 \cdot 0 + 0.6 \cdot 0 = 0
 \end{array}$$

$$\begin{array}{r}
 b \\
 0 \\
 1
 \end{array}
 \quad
 \begin{array}{r}
 V_2(1) \\
 0.4V_3(1+0) + 0.6V_3(1-0) = 0.4 \cdot 0 + 0.6 \cdot 0 = 0 \\
 0.4V_3(1+1) + 0.6V_3(1-1) = 0.4 \cdot 0.4 + 0.6 \cdot 0 = 0.16
 \end{array}$$

$$\begin{array}{r}
 b \\
 0 \\
 1 \\
 2
 \end{array}
 \quad
 \begin{array}{r}
 V_2(2) \\
 0.4V_3(2+0) + 0.6V_3(2-0) = 0.4 \cdot 0.4 + 0.6 \cdot 0.4 = 0.4 \\
 0.4V_3(2+1) + 0.6V_3(2-1) = 0.4 \cdot 0.4 + 0.6 \cdot 0 = 0.16 \\
 0.4V_3(2+2) + 0.6V_3(2-2) = 0.4 \cdot 1 + 0.6 \cdot 0 = 0.4
 \end{array}$$

$$\begin{array}{r}
 b \\
 0 \\
 1
 \end{array}
 \quad
 \begin{array}{r}
 V_2(3) \\
 0.4V_3(3+0) + 0.6V_3(3-0) = 0.4 \cdot 0.4 + 0.6 \cdot 0.4 = 0.4 \\
 0.4V_3(3+1) + 0.6V_3(3-1) = 0.4 \cdot 1 + 0.6 \cdot 0.4 = 0.64
 \end{array}$$

$$\begin{array}{r}
 b \\
 0
 \end{array}
 \quad
 \begin{array}{r}
 V_2(4) \\
 0.4V_3(4+0) + 0.6V_3(4-0) = 0.4 \cdot 1 + 0.6 \cdot 1 = 1
 \end{array}$$

Gambling game

b	$V_1(2)$	
0	$0.4V_2(2 + 0) + 0.6V_2(2 - 0)$	$= 0.4 \cdot 0.4 + 0.6 \cdot 0.4 = 0.4$
1	$0.4V_2(2 + 1) + 0.6V_2(2 - 1)$	$= 0.4 \cdot 0.64 + 0.6 \cdot 0.16 = 0.352$
2	$0.4V_2(2 + 2) + 0.6V_2(2 - 2)$	$= 0.4 \cdot 1 + 0.6 \cdot 0 = 0.4$

Curse of dimensionality

If s_t is multidimensional and/or continuous, enumerating all states is difficult or impossible. Instead of solving

$$\begin{aligned}
 V_t(s_{t-1}) &= \max_{c_t} r(s_{t-1}, c_t) + v \mathbb{E}_{W_t \bullet} [V_{t+1}(s_t) | s_{t-1}] \\
 \text{s.t. } s_t &= f(s_{t-1}, c_t, W_t)
 \end{aligned} \tag{13}$$

create sample paths (scenarios) $(W_{\bullet}^n)_{n=1}^N$ and solve iteratively for $n = 1, \dots, N$ and $t = 1, \dots, T$

$$\begin{aligned}
 \bar{V}_t^n(s_{t-1}^n) &= \max_{c_t} r(s_{t-1}^n, c_t) + v \mathbb{E}_{W_t \bullet} [\bar{V}_{t+1}^{n-1}(s_t) | s_{t-1}^n] \\
 \text{s.t. } s_t &= f(s_{t-1}^n, c_t, W_t^n),
 \end{aligned} \tag{14}$$

where \bar{V}_t^n is an approximated value function based on first n paths, see Powell (2009) for details.

Literature

- P. Kall, S.W. Wallace: **Stochastic Programming**, John Wiley & Sons, Chichester.
- W. B. Powell: **What you should know about approximate dynamic programming**, Naval Research Logistics 56(1): 239–249, 2009.
- W. B. Powell: **Perspectives of approximate dynamic programming**. Annals of Operations Research 241(1-2), 319–356, 2016.
- M. C. Sunny Wong: **Dynamic Optimization: An Introduction**, Lecture Notes – University of Houston, 2013.