MathDM PW4 - Stochastic Machine Replacement Problem

Yiping Fang, LGI, CentraleSupelec

November 30, 2021

Problem statement

At the end of each production cycle (e.g. seasonal) a candy production line must decide whether to keep a machinery again or replace it with a new one. A machinery at cycle t has a corresponding efficiency state $s_t \in S = \{1, 2, ..., 10\}$. We know the machinery's state at the first cycle $s_1 = 1$, and it has probability $p = 0.9$ to go to efficiency state $s_{t+1} = \min\{s_t+1, 10\}$ and probability $1 - p$ to go to efficiency state $s_{t+1} = \min\{s_t + 2, 10\}$ if not replaced by a new one. At each efficiency state s, it produces $y(s) = 8 + s - 0.15s^2$ tons of candy over the corresponding production cycle. We assume a machinery must be replaced upon completion of the production cycle $s_t = 10$ since it becomes too unproductive. The net cost of replacing a machine is $c = 500$ k€ and the profit contribution of candy is $m = 150$ k€ per ton.

This is an infinite horizon, stochastic model with time t measured in production cycles. The state is.

$$
\begin{aligned}
&s_t = \text{production efficiency state of a machine at production cycle } t\\
&s_t \in \{1, 2, ..., 10\}
\end{aligned}
$$

The action is

$$
\begin{aligned}
&a = \text{replacement decision}\\
&a \in A(s_t) = \begin{cases} \{\text{keep, replace}\} & s_t < 10 \\ \{\text{replace}\} & s_t = 10 \end{cases}
\end{aligned}
$$

The state transition function is

$$T(s_t, a, s_{t+1}) = \begin{cases} p & a = \text{keep}, s_{t+1} = s_t + 1, s_t \leq 8 \\ 1 - p & a = \text{keep}, s_{t+1} = s_t + 2, s_t \leq 8 \\ 1 & a = \text{keep}, s_{t+1} = 10, s_t = 9 \\ 1 & a = \text{replace}, s_{t+1} = 1, \forall s_t \end{cases}$$

The reward function is

$$r(s, a) = \begin{cases} y(s)m & a = \text{keep} \\ y(s)m - c & a = \text{replace} \end{cases}$$

Question: Write down the Bellman optimality equation of the value function. What replacement policy maximizes the expected long term cumulative profits? Using Value Iteration to solve the problem (you are also encouraged to use the Policy Iteration and Linear Programming method). Test the sensitivity of the optimal policies to different problem parameters, e.g., p and c.