STOCHASTIC OPTIMIZATION

Stochastic Machine Replacement Problem

Authors
Alexandre Robin
Antoine Poupon
Raphael Monges

1 Problem statement

Data

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. We know that:

- Replacing a machine costs 500 k€.
- The profit contribution of candy is 150 k€/per ton.
- The quantity of candies produced over a production cycle is given by the function $y(s) = 8 + s - 0.15s$ with s the efficiency state.

To solve this problem we will use a Markov Decision Process formulation. The reward function, the transition function and the action function are provided in the statement of the project.

The action function is

$$a = \text{replacement decision}$$

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

The state transition function is

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

The reward function is

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

Bellman optimality equation

The Bellman optimality equation is defined as:

$$V^*(s) \leftarrow \max_a \sum_{s'} T(s, a, s') \left[R(s, a, s') + \gamma V^*(s')\right]$$

As the states and actions spaces are finite, we know that if the discount factor is bounded between 0 and 1 and we update all the state value function a sufficient number of iterations, then we will converge to the optimal value function and consequently the optimal policy.

Therefore, we set $\gamma = 0.8$ and explore 2 solving methods which are the value iteration and the policy iteration algorithm.

2 Problem solving

Value iteration

We solve the problem using the value iteration algorithm. To this end, we initialize the value functions of all the states and then iteratively update them as follows :

$$V_{k+1}(s) \leftarrow \max_a \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma V_k(s')\right]$$

The algorithm converges in 46 iterations for a convergence criterion $\epsilon = 0.001$. We found that until the state 6 the optimal action is to keep the machine, then it becomes to replace the machine.

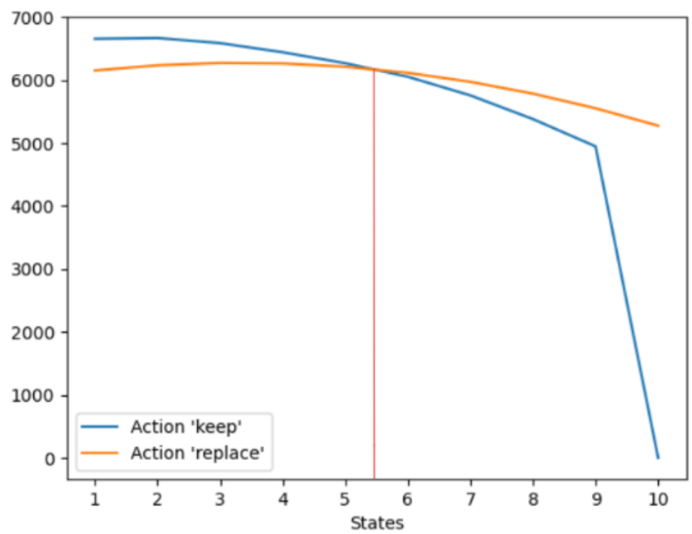

Figure 1: Action-value function when running the value iteration algorithm

We can see that for the state 5 the action-value function of the action 'keep' is still higher than the one of the action 'replace' which is not the case anymore for the state 6, so this is consistent with the output of the algorithm, given in the previous paragraph.

Policy iteration

We now solve the problem using the policy iteration algorithm. To that extent, we first compute the value of the policy iteratively :

$$V^{\pi}_{k+1}(s) \leftarrow \sum_{s'} T(s,\pi(s),s') \left[R(s,\pi(s),s') + \gamma V^{\pi}_k(s')\right]$$

Then we update the policy :

$$\pi_{k+1}(s) \leftarrow \arg\max_a \sum_{s' \in S} T(s,a,s') \left[R(s,a,s') + \gamma V^{\pi}_k(s')\right]$$

And we repeat this step until convergence. For the same convergence criterion ϵ as before, the algorithm converges in 3 steps and leads to the same results.

Linear programming

We check if a linear programming method leads to the same optimal policy as the 2 previous dynamic programming methods. Our formulation is the following one:

$$\begin{aligned}
&\min_{V \in \mathbb{R}^{|S|}} \quad \sum_{s \in S} V(s) \\
&\text{subject to:} \\
\text{for } s = \{1, \ldots, 8\}, \quad & V(s) \geq \max\left(y(s) \cdot m + \gamma\left(p \cdot V(\max(s-1, 10))\right.\right. \\
& \quad \left.\left. + (1-p) \cdot V(\min(s+2, 10))\right)\right) \\
\text{for } s = 9, \quad & V(s) \geq \max\left(y(9) \cdot m + \gamma V(10), y(10) \cdot m + \gamma V(1)\right) \\
\text{for } s = 10, \quad & V(s) \geq y(10) \cdot m - c + \gamma V(1)
\end{aligned}$$

Here again, we get the same results as before: replacing the machine becomes better than keeping the machine when the machines get to the state 6.

3 Sensitivity analysis

In this part, we test the sensitivity of the optimal policies to different problem parameters.

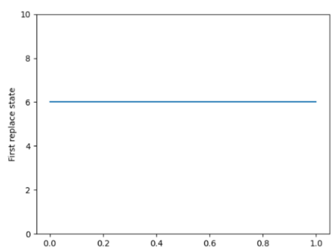

Figure 2: Parameter p

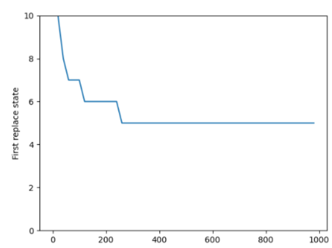

Figure 3: Parameter m

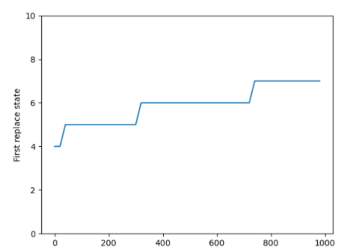

Figure 4: Parameter c

We can see that the probability p has no impact on when the first replace state occurs. Contrary to the probability p, changing the machine cost c and the profit contribution of candy m have impacts on when the first replace state occurs:

- The higher the machine cost c is, the later it is worth to replace machines.
- The higher the profit contribution of candy m is, the earlier it is worth to replace machines. Though, above a certain amount m, the profit contribution of candy has no impact anymore on when to replace machines.