Dynamic Programming Project

Vincent Leclère

Work to be sent by email to vincent.leclere@enpc.fr before 28th of April 2023. Can be done in pairs. The preferred format is jupyter notebook (ipynb and pdf version), code + pdf also accepted. Code should be commented and readable.
Expected total work time: 20 hours.

1 Maintenance problem

1. We consider a company that operates numerous vending machines. They are changed every year (12 months). A machine can be new, in good shape, old or broken. Each month you can either do nothing (cost 0), maintain (cost 10), repair (cost 15 if in good shape, 30 if old, 50 if broken) or replace (cost 70) a machine. A repaired machine will be in good shape at the end of the month. A replaced machine will be new at the end of the month. No action can be taken on a new machine and it will be in good shape at the end of the month. Without maintenance (resp. with maintenance), a machine in good shape will be in good shape at the end of the month with probability 0.3 (resp. 0.8) and old with probability 0.7 (resp. 0.2). Without maintenance (resp. with maintenance), a machine in old shape will be in old shape at the end of the month with probability 0.5 (resp. 0.9) and broken with probability 0.5 (resp. 0.1). A broken machine will stay broken unless it is repaired or replaced.

   A new machine earn 30, an in good shape machine earns 20, an old earn 10 and a broken machine earn 0 for the month.

   (a) Justify why maximizing the expected revenue makes sense.

   (b) Describe the problem in terms of a controlled Markov Chain.

   (c) Solve the problem using Dynamic Programming.

2 Stock management

We consider a shop that has a stock of a single product that should stay between 1 and 20. Each evening he can order for the next morning up to 5 products (current stock + order cannot exceed 20), with a price of 1. Having a stock x at the end of the day cost $0.1x$ to the owner. During the day t he can sell products up to d_t for a price of 3, where d_t is a random demand. Unmet demand is lost. After $T = 14$ the stock is lost. We start with an initial stock of 10 products.

The demands d_t are stochastically independent binomials of parameter (p_t, n) where p = [0.2 0.2 0.4 0.4 0.7 0.7 0.2 0.2 0.8 0.8 0.5 0.5 0.2 0.2]

2. Simple stock problem

   (a) Describe the problem in terms of a stochastic control problem. What is the state? What is the control? What is the cost? What is the dynamics?

   (b) Write the Bellman equation for this problem.

   (c) What is a policy for this problem? Write a simulator taking as argument a policy and an integer returning the estimated expected cost associated with the policy with 95% confidence interval.

   (d) Suggest a policy and simulate it. What is the expected cost of this policy?

   (e) Solve the problem using Dynamic Programming.

   (f) Simulate the optimal policy and check that the optimal value is indeed obtained.

3. Advanced stock problem
   (a) Using the code you wrote for the previous question, solve the problem over $T = 96$ days (with periodic demand).
   (b) We now assume that the order passed at the end of day t is received at the beginning of day $t + 2$. Write a dynamic programming equation for this problem (with horizon $T = 14$). What is the new optimal value?

3 Dice trading

We consider a game of dice. At the beginning of a turn, the player has a certain amount of points (starting with 0). He can decide, if he has at least 6 points, to buy one new die (one per turn maximum, to be kept until the end) for 5 points. The player then throws his dice (6 faces, independent, equilibrated), and add the maximum of all dice to his points. The game plays for $T = 10$ turns. The player wants to maximize the expected number of points at the end. For simplicity, we assume that we can have at most 3 dice.

Example (10 points, 4 turn game) :

turn	dice roll	action	total points
1	3	can't buy	0+3=3
2	4	can't buy	3+4=7
3	5	don't buy	7+5=12
4	max(3,2)	buy	12-5+3=10

4. Simple game
   (a) Determine the dynamical system considered in this problem. (specify state, control, dynamics)
   (b) A strategy is a function taking as argument the time-step t and the current state x and returning a control. Implement a very simple strategy (a heuristic).
   (c) Write a simulator taking as argument a strategy and an integer returning the estimated expected cost associated with the strategy with 95% confidence interval.
   (d) Compute the law of the maximum of 1,2 or 3 dice
   (e) Find the optimal value V_0 and strategy π^* by Dynamic Programming. Describe the optimal strategy in simple terms.
   (f) Check by simulation that the optimal value V_0 is indeed obtained when using the strategy π^*
   (g) For which horizon T will it never be interesting to buy?
   (h) What happens if we do not restrict the maximum number of dice that one can have? (still buying only one per turn)
5. Spending dices.

   We now consider an extension of the previous game. At any turn, once the dice are thrown, the player, if he has at least 2 dice, can spend a die to double the gain of the throw. As before he can buy a die at the beginning of the next round.
   (a) Assuming that we can have at most 5 dice, compute the optimal value and describe the optimal strategy.
   (b) What happens if we do not restrict the maximum number of dice that can be owned by a player?
   (c) Returning with a maximum of 5 dice, we now assume that, at the end of the game, the remaining dice are sold to the "next player" for K where K is given in the following table What is the

D	1	2	3	4	5
K	0	2	4	5	8

   optimal value of this new problem?
   (d) What value K should you use at the end of the 10 turn game to represent a 20 turn game?
   (e) Suggest an efficient way of finding a quasi-optimal strategy and value for a 10^{10} turns game.