

MS&E 214: HW 4, Spring 2023-24

This HW should not be submitted. It is meant as a set of practice problems to help you prepare for the final exam, and covers material that is part of the class but was not covered by the previous HWs.

- (1) You have N identical units of a good (e.g. seats for the same class of travel on a flight) that you want to sell over T days (e.g. the flight may be T days in the future). You are given an “oracle” that can tell you, given any number K , day t , and price p , the probability $q(K, t, p)$ that exactly K customers will be willing to buy this good on day t if the price on day t is p . Assume that demands on different days are independent.
 - (a) You are allowed to set a different price each day, out of M possible prices $p_1, p_2, \dots, p_M$ and your goal is to maximize the expected revenue. Present a dynamic program to compute this maximum revenue, and estimate the number of steps required to solve this. Clearly write down your state space. Solve this using a dictionary to store the values that you have already calculated, as described in class, or an array, for $N = 100$, $M=10$, and $T = 10$.
 - (b) What does your dynamic program actually output? How will you modify it to also compute the optimum price?
 - (c) What can you do to minimize making many deep recursive calls?
 - (d) How would you modify your solution if the prices were required to be non-decreasing each day?
- (2) Numerically solve the 2x2 adwords problem discussed in class. Compute the additional revenue the search engine would make if the budget of each advertiser went up by one cent. Also compute the additional revenue the search engine would make if the number of impressions of each keyword went up by 1.
- (3) ARC Consultants need to know how long it will take to complete a complex job. The job consists of N separate tasks. Task k takes t_k hours to complete once it gets started (regardless of how many people work on it). Additionally, certain tasks cannot be started until other tasks have been completed. This information is given to you in an $N \times N$ matrix P , where $P(A, B)$ is 1 if task B must be completed before task A can be started, and 0 otherwise.

Note that all elements on the diagonal must be zero as a task cannot be performed before itself. There is no limit on the number of tasks that can be performed at once.

 - (a) Formulate this as a shortest path problem [*Hint: : you need to use negative weight edges*]

- (b) Draw the shortest path formulation when P is as given below and the times for the tasks are given by $t_1=2$, $t_2=3$, $t_3=1$, $t_4=4$, $t_5=2$. Clearly show the nodes, edges, edge costs, and the source and the destination. Also, mark the shortest path on the graph.

$P(A,B)$	$B=1$	2	3	4	5
$A = 1$	0	0	1	1	0
2	0	0	0	0	1
3	0	0	0	1	0
4	0	1	0	0	1
5	0	0	0	0	0

- (4) You are given two (α, β) machines: (1,2) and (10000, 10000). Find which of these you would play on the first day for values of $\theta = 0.8$ and 0.95. Feel free to use the spreadsheet from class as your starting point:

https://web.stanford.edu/~ashishg/msande214/handouts/gittins_solved.xlsx

- (5) (Advanced) Complete the proof of the optimality of base stock policies for the simple setting described in class, by considering the case $a > b$ (slide 11 of https://docs.google.com/presentation/d/1cdu6NHsDn_-T7An5aZHA9Zik919o1Hrh-ZhWqyYFHqc)

Note: Review the solutions but don't worry if it is not completely transparent or if you are not confident you could have come up with the proof in the context of an exam; this problem is too hard to be in the exam without additional hints.

- (6) (Advanced) Consider a currency exchange where you can change any of M currencies into any other currency. You are given a rate $R(i,j)$ for each currency pair, i.e. you will get $R(i,j)$ units of currency j for 1 unit of currency i . Write a network flow problem that will help you determine whether there is an arbitrage opportunity in this network.