Optimization and Uncertainty

Summer term 2023

Prof. Dr. Martin Hoefer
Conrad Schecker, Lisa Wilhelmi

GOETHE UNIVERSITÄT FRANKFURT AM MAIN
Institute of Computer Science
Algorithms and Complexity

Assignment 6

Issued: 30.05.2023
Due: 13.06.2023, **10:00h**

This assignment is due on **13 June**, i.e., **two weeks** after publication.

Exercise 6.1 *Stochastic* KNAPSACK (4 + 4 points)

In the Stochastic Knapsack problem, like in the non-stochastic variant, items of values v_i and sizes s_i are packed into a knapsack of a fixed capacity c; the goal is to maximize the sum of values of packed items. In the stochastic variant, the sizes are random. The size of an item is revealed only after it has been packed into the knapsack. If it exceeds the knapsack capacity, the item is removed from the knapsack and no further items can be packed. Again, optimal policies are adaptive, meaning that their choices depend on the sizes of already packed items.

a) Show that every policy corresponds to a solution of the following linear program:

$$\begin{aligned} \text{maximize} \sum_i v_i x_i & \\ \text{subject to} \sum_i x_i \cdot \mathbb{E}[\min\{s_i, c\}] &\le 2c \\ 0 \le x_i \le 1 \text{ for all } & \ i. \end{aligned}$$

Hint: Construct a sequence of random variables $Y_{t\in\mathbb{N}}$ such that $\mathbb{E}[Y_{t+1} \mid Y_t] = Y_t$. Assume that, in this case, $\mathbb{E}[Y_t] = \mathbb{E}[Y_0] = 0$ for any $t \ge 0$.

b) For simplicity, assume that $s_i \le c/2$ for all i with probability 1. Now, use an optimal LP solution x^* to determine a randomized non-adaptive policy as follows: Pack item i into the knapsack with probability $x_i^*/8$. Show that this policy achieves an expected reward of at least $\frac{1}{16}\sum_i v_i x_i^*$.

Exercise 6.2 *Adaptivity gap for* k-TESTMAX (4 points)

Design a non-adaptive algorithm for k-TESTMAX in the IID scenario and show that the adaptivity gap is $\mathcal{O}(\log \min(n, k))$.
Hint: First derive the probability of finding a good item. Then choose the number of tests per item accordingly.

Exercise 6.3 *Fair cap* (2 points)

Consider the following distribution for the prize of box i: the prize v_i is equal to w_i with probability q_i and is 0 else. Compute the fair cap.

Exercise 6.4 *Pandora Box as MDP* (2 Points)

Formulate the Pandora Box problem as Markov Decision Process. Explain your modeling choices briefly.

Exercise 6.5 *Fixed Order Pandora Box* (3 Points)

In the Ordered Pandora Box problem boxes have to be opened sequentially in an arbitrary, fixed order. The order is given in advance. At each step, we can either invest the opening cost of the next box (according to the given order) or stop the process (and select the best reward in any opened box).

Suppose all boxes have the same support of size m, but possibly different probabilities for each realization (i.e., the set of possible prizes is the same for all boxes and contains m elements). Prove that an optimal policy for Ordered Pandora Box can be computed by dynamic programming in time $\mathcal{O}(nm^2)$.

Exercise 6.6 *Pandora Box Matching* (4 points)

In order to generalize the Pandora Box setup from the lecture, suppose the task is to match people $i \in [n]$ to boxes $j \in [m]$, where each person can take home at most one prize. Person i's value v_{ij} for the prize in box j is independently drawn from a distribution D_{ij}, but it costs c_{ij} to learn the exact value v_{ij}. Consider $A_{ij}, I_{ij}, \omega_{ij}, \kappa_{ij}$, and b_{ij} to be the corresponding generalizations of the variables introduced in the lecture. Show that for any policy π, the expected reward R is given by

$$R(\pi) = \sum_{i \in [n], j \in [m]} \mathbb{E}\left[A_{ij} \cdot \kappa_{ij} - (I_{ij} - A_{ij}) \cdot b_{ij}\right].$$