# Optimization and Uncertainty

Summer term 2021

Prof. Dr. Martin Hoefer
Tim Koglin, Lisa Wilhelmi

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

# Assignment 3

Issued: 04.05.2021
Due: 11.05.2021, **10:00h**

**Exercise 3.1** Secretary *with multiple jobs* (4 points)

Let $k \in \mathbb{N}_{>0}$. Design an online algorithm that accepts the $k$ best candidates in the Secretary problem with constant probability. Assume that the total number of candidates $n$ is at least $2k$. Depict your algorithm and show that the probability to accept the $k$ best candidates is constant for any fixed $k$.

**Exercise 3.2** Prophet *with multiple items* ($4 + 4 + 2$ points)

For the Prophet problem, let $p_i$ be the probability that $v_i$ is optimal, where $i = 1, ..., n$. Furthermore, suppose that $\tau_i$ is such that $\Pr[v_i \geq \tau_i] = p_i$, i.e., the $p_i^{\text{th}}$ percentile for $v_i$. For simplicity, assume that such a threshold $\tau_i$ always exists. Define

$$\tilde{v}_i(p_i) := \mathbb{E}[v_i \mid v_i \geq \tau_i]$$

as the expected value of $v_i$ given that it lies in the top $p_i^{\text{th}}$ percentile. We consider the following strategy to pick one item: *When item $i$ shows up, if no item has been chosen among $1, ..., i-1$, reject it with probability 1/2 outright, else accept it if $v_i \geq \tau_i$.*

a) Show that the algorithm achieves a value of at least $\frac{1}{4} \cdot \mathbb{E}[v_{\max}]$.

b) Suppose we are now allowed to choose $k$ out of the $n$ items. Then, the quantity of interest is given by $\mathbb{E}[\text{sum of values of best } k \text{ items}]$. Show that the algorithm achieves at least $\frac{1}{4}$ of it.
*Hint: Redefine $p_i$ as the probability that $v_i$ is among the top $k$ values.*

c) If the distributions are finite, then a threshold $\tau_i$ as defined above might not exist. Can you adjust the algorithm to handle this case? Explain your answer.

**Exercise 3.3** *NP-Hardness of offline* Item Allocation (4 points)

Show that the offline version of the Item Allocation problem is NP-hard.
*Hint: Use* Independent Set *for the reduction.*