Research Proposal

Tim Grimbergen, Jesse Flikweert, Agostino Sorbo, Daan van Dongen and Timo Post

09-29-2023

In this research we will investigate the online ticket-hotel problem. In this problem there are n people that have to be flown home in m days. For each day $i \in \{1, \ldots, m\}$ the algorithm receives the number of available plane tickets s_i, the price of a ticket p_i and the hotel price h_i. Every person that is not sent home by plane must spend the night in a hotel. The algorithm must then decide irrevocably how many people it will sent home f_i on the given day.
In this project we will specifically be investigating the problem under the following constraints

1. $n \in \mathbb{N}$,[1]
2. $m \in \mathbb{N}$,
3. $s_i = n$ for all $i \in \{1, \ldots, m\}$,
4. (a) $p_i \in \mathbb{N}$ OR,
   (b) $p_i \in \{1, 2, \ldots, p_{\max}\}$ with $p_{\max} \in \mathbb{N}$,
5. $h_i = \bar{h}$ with $\bar{h} \in \mathbb{Z}_{\geq 0}$ a constant.

Note that for the fourth constraint we decide between conditions a) and b). Any online algorithm knows these constraints. So, for a given instance it knows $n, m, p_{\max}$ and that $s_i = n$ for all i. Our project then consists of three objectives.

Goal 1: No deterministic competitive algorithm for 4.a)

We want to prove that with constraint a) the problem does not admit any competitive deterministic online algorithm. Specifically, we prove the following proposition.
Proposition 1: *There does not exist an algorithm* ALG *that is K-competitive for some real number $K \in \mathbb{R}$.*
The proof strategy is to design for any algorithm ALG test instance I so that $\text{ALG}(I) > K \cdot \text{OPT}(I)$.
A paper that has proven a similar result for the multi-core paging problem is provided here.

Goal 2: A deterministic algorithm and its competitive ratio for 4.b)

From this point on, we will only be working with constraint 4.b) since we found that 4.a) cannot yield interesting results. For simplicity we start with the case $\bar{h} = 0$ and then move on to the more interesting case $\bar{h} > 0$.
For example for the case $\bar{h} = 0$ we might consider the algorithm that for $i < m$ buys n tickets on day i if and only if $p_i \leq \lfloor \sqrt{p_{\max}} \rfloor$ and for $i = m$ always buys n tickets if there are still people that have to be sent home. We conjecture that this algorithm is $\sqrt{p_{\max}}$-competitive. Additionally, we might prove that this algorithm is optimal.

[1] In this research $\mathbb{N}$ is the set of all strictly positive integers $1, 2, \ldots$.

Goal 3: A randomized algorithm and its competitive ratio

Given constraints and algorithms for Goal 2, we'll adjust those algorithms to have their decision statements contain an additional, randomized variable. For example, for Goal 2, we might not just deterministically make a decision based on $p_i \leq \lfloor \sqrt{p_{\max}} \rfloor$, but on $p_i \leq \lfloor (1 + \lambda_i) \sqrt{p_{\max}} \rfloor$, with $\lambda_i \in \mathcal{U}(-\lambda, \lambda)$ being randomly sampled for some real number λ. This allows us to compare average results of experiments for different values of λ. Also, we might derive theoretical results for the competitive ratio c, where c is such that

$$\mathbb{E}(\mathrm{ALG}(I)) \leq c \cdot \mathrm{OPT}(I),$$

for all instances I (see Eq.2 in this paper).

Goal 4: Experimental analysis of all proposed algorithms

Compare the theoretical competitive ratios with averaged results on a generated set of instances (see for example figure 1 in this paper).
The first step towards this goal is to generate the experimental instances. These instances will be randomly generated according to the aforementioned problem constraints. The aim of the experiment is to compare the competitive ratios of:

- the deterministic algorithms
- the randomized algorithm

The experimental ratios will be averaged across all the experiment instances. The theoretical competitive ratios are then plotted against the experimental ones. This can be done comparably to how the authors of this paper have done on pages 6 and 7.