ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

Prof. R. Wattenhofer

BA:

The On-Line k-Taxi Problem

Consider a traffic net of a city. A taxi company wants to match their drivers to the customer requests that appear everywhere in the city. A customer request contains the information, where the taxi can pick up the customer and the customer's destination. An algorithm should decide which driver picks up which customer.

The algorithm should minimize certain criteria: A customer becomes unsatisfied fast, so the customer's waiting time is an important factor that should be considered. The company wants to gain as much profit as possible, so the taxi driver's waiting time is another criterion. In reality the problem is often generalized to more complex space by taking into account additional parameters such as the size of a taxi and the number of customers, the working hours of a taxi driver or statistical data of earlier customer requests. Additionally, at any point in time there could appear a new customer on the map, who is closer to a taxi than the customer that the taxi is matched to.
The naive approach that matches the customers immediately to free taxis does a terrible job. The free taxi could be on the other side of the city. Another taxi might finish a job right in the area of the customer or a new customer could appear who fits better to the free taxi. To handle this task, we must allow the algorithm to delay its service.
In this project, we need to device competitive algorithms and/or show hardness of this problem. It would be interesting to consider variations of the problem and related problems.

Contacts

- Georg Bachmeier: georg.bachmeier@tik.ee.ethz.ch, ETZ G94
- Yuyi Wang: yuyi.wang@tik.ee.ethz.ch, ETZ G94

Detailed Project Outline

We denote the following primary tasks mandatory (on the right side you find a rough estimate for the time that we allocate to the respective task):

- Collect related work from publications about the k-taxi and similar problems. Extract information about previously successful algorithms. Model the task of the algorithm and formally define the input and output. Define a suitable cost function for the algorithm. ($\star$)
- Analyze the optimal solution for small subsets of the problem, for example with restricted number of taxis, customers or nodes in the graph. Use a previously successful algorithm to solve these subsets or prove that they can not be applied. If necessary develop a custom algorithm that can handle these small cases. ($\star\star\star\star$)
- Generalize the algorithm from ii) to the problem without the restrictions. Improve the algorithm until it is possible to prove that the algorithm still outputs cost efficient results. ($\star\star\star\star$)
- Analyze the resulting algorithm. Show the boundaries and outline how the algorithm can be improved to be more cost efficient or to support a larger input space. ($\star\star\star$)

Extensions

Apart from these requirements, we can think of plenty of ways to extend this thesis:

- Discuss variations of the original problem and how the algorithm can be adapted to produce cost efficient results for these variations.
- Develop a software simulation of the problem and include the implementation of the algorithm. It should be possible to configure the simulation to different corner cases of the problem.