Karlsruher Institut für Technologie
Fakultät für Informatik

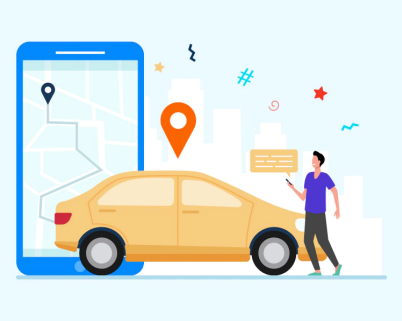
Lehrstuhl Algorithm Engineering
Prof. Dr. Peter Sanders

Bachelor-/Masterarbeit

Parallel Dynamic Ridesharing

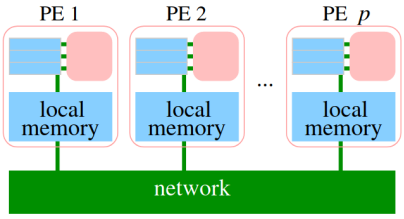
Description

Increasingly, calls for more sustainable living include the transformation of personal traffic from individual transportation (e.g. cars) to shared transportation (e.g. public transit, carpooling). One promising approach for this is *ridesharing*, where individuals hitch a ride in a vehicle which they may share with other passengers that are traveling in a similar direction. Ridesharing is often faster than public transit but more cost and space efficient than individual cars.



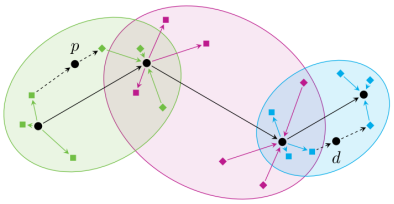
In *dynamic ridesharing* systems, a user issues a request to travel from their current location to a destination. A dispatching algorithm then matches this request to a vehicle which picks up the passenger at their location after a certain wait time. The dispatching algorithm usually attempts to assign a request to a vehicle s.t. the vehicle's detour for serving the request is minimized while adhering to a number of constraints on the maximum wait time and trip time of the passenger.

Since the demand for ridesharing is increasing, it is vital to have efficient dispatching algorithms that can handle a large number of vehicles and a high frequency of requests. Recently, advancements have been made in this area on the basis of algorithmically engineered shortest path algorithms. The sequential LOUD algorithm[1] outperforms existing competitors by an order of magnitude using contraction hierarchies (CHs) and bucket based many-to-many shortest path algorithms.
We would like to improve on these advancements further by parallelizing the job of dispatching requests to vehicles. With a parallel implementation, issues like the consistency of the system state (vehicle routes and stops) or the quality of matchings between requests and vehicles arise.



Topic of this work

It is the goal of this work to create a parallel dispatching algorithm for the dynamic ridesharing problem. An implementation of this parallel algorithm is to be produced in C++ on the basis of the existing LOUD algorithm. Both the quality and efficiency of the resulting algorithm is to be experimentally compared with existing algorithms, mainly the sequential LOUD algorithm. Furthermore, the scalability of the algorithm needs to be evaluated on large scale ridesharing scenarios which may require generating such scenarios by synthetically upscaling smaller benchmark instances.



Prerequisites

- Interest in and a base knowledge of parallel algorithms, preferably also of route planning algorithms
- Good programming knowledge in (modern) C++, preferably also in parallelization interfaces like MPI

[1] Buchhold, Valentin, Peter Sanders, and Dorothea Wagner. FFast, Exact and Scalable Dynamic Ridesharing."2021 Proceedings of the Workshop on Algorithm Engineering and Experiments (ALENEX). Society for Industrial and Applied Mathematics, 2021.

Moritz Laupichler
Informatikgebäude am Fasanengarten, Raum 207, moritz.laupichler@kit.edu