

Task: Collaborative Carrier Network

PBL in Software Development

Task Description:

- Carriers want to optimize their profit of their daily “pickup and delivery“ transport business
- The marginal profit of carries currently decrease. Therefore, carriers have a strong interest to cooperate with other carriers to optimize their profit.
- A system has to be implemented that supports the reassignment of transport requests. The solution of the reassignment should not decrease the profit of agents.
- An auction-based solution should be used to support privacy of agents.
- A graphical use interface should support the analytical view of solutions.

Subtask: An independent task is to ensure that the software components are always tested well.

ST01: Development

GitLab of TUHH is used for the complete development process to support future maintenance issues. The project is at source level. At the end, the product owner will take over the infrastructure. As the product owner, I want to have access to all development stages. Setup the infrastructure so that at the end of each Sprint the product owner can download the resulting product including scripts to start the application.

ST02: Agent Infrastructure

An agent platform has to be chosen. Two kind of agents must be implemented. Carrier agents and one Auctioneer agent. Carrier agents can register for auctions and can communicate with Auctioneer agents.

ST03: Tour planning

On start, each agent has a set of transport requests. An optimal tour has to be found to execute the transport period (one day). Initially a carrier has only one vehicle. The tour is visualized to the owner of the agent. Therefore, an agent GUI is needed.

ST04: Selecting transport requests for an auction.

The carrier agent have a price model for a transport request. The price has to be paid by the shipper. The agent also has an internal cost model for transport requests.

Based on this model he can calculate the cost, the profit of a tour and the real cost for a transport request. Transport requests that have a profit less than some threshold will be selected for an auction. Those requests are send to the auctioneer.

ST05: Simple auctions

The Auctioneer starts an auction. He selects one transport request and ask the carriers to bid. The carriers have to determine the worth of the transport request and send their bids to the auctioneer. This is repeated until all transport requests are sold or the limit of iterations is reached. Transport requests that can't be sold, are returned to the original carrier. Different auction mechanisms are possible. Initially a vickery auction should be used. The auctioneer agent has a GUI that shows the complete process.

ST06: Evaluation

After the end of an auction, the results should be visualized to be able to compare the goodness of the solution and the initial situation for each carrier. The GUI of the agent shows the results.

ST07: Auctions with bundles

The simple auctions should be extended so that the carriers can also bid for bundles of transport requests. The auctioneer must send the possible bundles to the carriers. The carriers have to bid for the bundles. Afterwards, the auctioneer hast to compute the best distribution of bundles to carriers. Different bundles might have the same transport request. The solution must guarantee that a transport

request is not contained in two bundles.

ST08: how to integrate this in the overall development process (GitLab, CI)

- You should integrate your testing environment with CI so that tests can be executed when new code is deployed.