Negotiation in Ride-hailing between Cooperating BDI Agents

Ömer Ibrahim Erduran*[a], Marcel Mauri*[b] and Mirjam Minor*[c]
Department of Computer Science, Goethe University, Frankfurt am Main, Germany

Keywords: Multi-agent Cooperation, Environmental Sustainability, Distributed Problem Solving, Ride-hailing, Mobility as a Service, BDI Architecture.

Abstract: These days, ride-hailing is an emerging trend in Mobility as a service (MaaS). First services involving human taxi drivers such as Uber, Lyft and DiDi are commercially successful. With the rise of autonomous vehicles, self-organized fleets for ride-hailing systems come into the focus of research. Multi-agent systems (MAS) provide solutions for many challenges of this application scenario. Especially, the communication of cooperating agents is beneficial for a structured and well planned task distribution. In this paper, we investigate a MAS for autonomous vehicles in MaaS and put the focus on a negotiation based assignment of customer trips. An agent model concept is introduced where the main type, the *vehicle agent* is designed following a BDI architecture. The communication system for the MAS is implemented by using the *contract net protocol*. We develop the negotiation process and furthermore evaluate the agent communication with respect to its impact on pickup time satisfaction and environmental sustainability using two quality measures, which calculate the average travel distance and the order dropout rate. An experimental setup including historical trip data in a simulation demonstrates the feasibility of our approach.

1 INTRODUCTION

Due to frequent traffic congestions and problems with parking space in urban areas, city residents tend to refrain from owning private cars (Pavone, 2015). Mobility as a service (MaaS) *"stands for buying mobility services as packages based on consumers' needs instead of buying the means of transport"* (Kamargianni et al., 2016). E-mobility with autonomously driving vehicles will contribute to this ambitious goal. Today, however, there are still many open issues in this field and MAS are capable of providing novel solutions for MaaS applications. This paper addresses a real-world scenario of *ride-hailing* using multi-agent cooperation and negotiation for a fleet of autonomously driving vehicles. In ride-hailing, a single passenger is served by a single autonomous vehicle (Qin et al., 2020). The main task of the MAS is to provide trip services for customers, who can call an autonomous vehicle, for this purpose on demand. We address in our work the self-organizing management which has its roots in distributed problem solving by considering trip requests with negotiating agents acting as a single fleet. We further compare this method with a centralized approach where the trip processing takes place in a greedy manner. One aspect that needs to be considered in mobility scenarios is its impact to the environment. The distribution of customer trips as tasks among the fleet leads to reduction of power consumption through less driven distances. We propose a novel domain specific concept of a multi-agent model with different types of agents, an application of the contract net protocol (CNP) for the communication between the BDI agents representing the autonomous vehicles. The core contribution of this paper is to investigate the benefits of negotiation in a decentralized MAS for the given scenario of autonomous taxi fleet. As a proof of concept, we measure the improvement in terms of missed trips and battery consumption while letting a fleet operate in our experiments. The latter is directly derived from the overall travel distance to serve the customer requests. The research question leads to the formulation of the following hypothesis:

A utility based negotiation solution for delegating trip assignments among a fleet of autonomous vehicle agents will reduce the rate of missed customer trips as well as decrease the energy consumption.

[a] https://orcid.org/0000-0002-1586-0228
[b] https://orcid.org/0000-0002-4135-1945
[c] https://orcid.org/0000-0002-6592-631X
*The authors have contributed equally to this work.

Erduran, Ö., Mauri, M. and Minor, M.
Negotiation in Ride-hailing between Cooperating BDI Agents.
DOI: 10.5220/0010973700003116
In *Proceedings of the 14th International Conference on Agents and Artificial Intelligence (ICAART 2022) - Volume 1*, pages 425-432
ISBN: 978-989-758-547-0; ISSN: 2184-433X

The remainder of this paper is structured as follows: In Section 2, we discuss related work considering research topics of autonomous vehicles for trip services. Section 3 deals with the design of the MAS with different agent types containing the architecture of the vehicle agents as well as the state model for the trip request processing. The trip assignment process of the MAS during negotiation is evaluated in Section 4.1. In Section 5, we discuss future work and conclude with a summary.

2 RELATED WORK

The research area for this work is twofold: On the one side, we discuss existing work studying the multi-agent approach of trip assignments for vehicle sharing systems. On the other side, we investigate recent applications and works which consider autonomous vehicles as sharing systems in e-mobility.

Applying the BDI agent architecture (Silva et al., 2020) for vehicle agents has been investigated recently in (Rüb and Dunin-Keplicz, 2020), which tends to be the most relatable work to our approach. The authors realize traffic agents considering a subsumption architecture in the agent design step and focus on small-scale traffic. The difference to our agent architecture is that the work mainly focuses on the agent design and implementation process without an evaluation of the performance. However, in our work we investigate the fleet performance concerning negotiation with critical measurements which are typical in the ride-hailing scenario (e.g. *estimated time of arrival* (ETA) (Fu et al., 2020)). Experimental results are introduced in (Certicky et al., 2014), where the authors study the travelled distance and success rate with the support of a simulation. Further results are shown in (Jaroslaw Kozlak and Zabinska, 2013), where the agent simulation is processed with JADE, which is also considered in our work. The focus in both works lies on a setting similar to the ride-hailing scenario. However, they both do not use the BDI architecture for their vehicles. In contrast, our work focuses on the efficient distribution of trip requests so that the vehicle fleet takes advantage of negotiation considering its own utility. Moreover, we consider free-floating data, which reflects the property of trips carried out by autonomous vehicles. The research problem addressed in (Malas et al., 2016) is similar to ours but here, neither BDI agents are considered nor the CNP is used for trip assignment. The specific usage of BDI agents is considered in (Deljoo, 2017) where the agents plan their work based on their utility function. As mentioned before, we set the focus on negotiation, which is significant for the MAS during processing and not explicitly on agent planning. Multi-agent approaches in context of traffic scenarios are discussed in (Bazzan and Klügl, 2014).

The considered application system is also well known as *Autonomous Mobility on Demand (AMoD)*, which relates to a mobility type, where autonomous vehicles provide transportation services for customers (Pavone, 2015). In (Pavone, 2015), the authors present a centralized stochastic solution using a spatial queuing model for the trip assignment problem, which differs from our multi-agent approach. They focus on an optimal routing of high-scale AMoD systems as well as their economical viability and acceptability in society. In (Danassis et al., 2019), a heuristic-based approach for solving the trip assignment problem is presented. The decentral nature of the proposed heuristic leads to no communication between participants as well as a high scalability. For their computation of trip assignment matching, they use a deviant utility function for the trip assignment process considering only the time factor. Furthermore, they optimize the assignment problem with linear programming, whereas we use distributed problem solving by means of cognitive agents and negotiation. The usage of CNP is investigated in (Yu and Zhang, 2010), where the scheduling of truck vehicles is developed in a pickup and delivery scenario. The evaluation therefore delivers results concerning the scheduling, whereas in our work, we use the CNP for the trip vehicle assignment.

3 DESIGN OF THE MAS

The main task of the MAS is the collaborative management of trip requests and battery power. An agent model with different agent types has been designed for this task. Currently, the agents are running in a simulation environment. In future, the agents are supposed to run as well in a real environment with sensory inputs from autonomously driving vehicles. The vehicles will also serve as actuators for physical movements in addition to a communication system between the agents.

3.1 Agent Model and Communication

The agent model comprises the following agent types: a *vehicle agent* represents an autonomous vehicle within a fleet, an *area agent* represents a section of the outdoor grounds where the vehicles operate, a *taxi office agent* represents the interface to the customers for a larger territory. The taxi office agent serves as a preliminary broker for the incoming customer requests. The agent uses a *zone model* for the area agents in its entire territory and also provides a registry service for the vehicle agents. The zone model determines the area agent in whose zone the customer has sent the trip request. Further, it knows approximately where the vehicle agents are since they register and de-register themselves at the area agents when entering or leaving a zone. From time to time, the area agents inform the taxi office agent on the number of registered vehicle agents. The taxi office agent forwards incoming customer requests to one of the vehicle-populated zones that are in close proximity to the customer request. The registration data has a higher time accuracy than in the zone model of the taxi office agent. The area agent initially assigns incoming trip requests to an agent within its zone and might select the agents arbitrarily according to their recent location, in rotating order, or just choosing the most recently registered agent. In addition to the registry and pre-assignment service, the area agent provides a *neighborhood service* that determines the vehicle agents in the proximity of the vehicle agent which is searching for negotiation partners. The latter reduces the communication load during the negotiation. The vehicle agent decides for each trip request whether it tries do delegate it or whether it takes over the trip itself. It is responsible for scheduling all trips to which it has committed itself, including trips for battery recharging. The delegation process follows the CNP (Smith, 1980) for agent negotiation. A vehicle agent that aims to delegate a trip request takes the role of a manager announcing the trip request to be negotiated on. The neighboring vehicle agents take the role of a contractor bidding for the trip request. The manager evaluates the bids and awards a contract to the bidder it determines to be most appropriate.

3.2 Agent Architecture

All agent types of the agent model follow a cyclic infinite sequence of iterations of the observe-think-act agent cycle (Kowalski and Sadri, 1999). The taxi office agent and the area agents adopt a stimulus-response architecture (Kowalski and Sadri, 1996), which means that the decision is a direct response to a sensory input, for instance to an incoming message. Thus, the agent shows a reactive behavior. The vehicle agents are designed in the BDI agent architecture (Rao and Georgeff, 1995), which is a more sophisticated specialisation of the observe-think-act cycle (Kowalski and Sadri, 1999) with a richer 'think' phase than stimulus-response. *B* stands for beliefs and represents the agent's assumptions on its own, internal state and the state of the environment. A sample belief of a vehicle agent is the current battery charge level or the own location in the environment. *D* stands for desires and represents the agent's objectives to be accomplished. An example desire is to fulfill a new trip request within reasonable time. It might be in conflict to other desires, such as maintaining a certain level of battery charge or serving already committed trip requests. *I* stands for intentions and represents the currently chosen course of action. A sample intention is to delegate a trip request by negotiation to another agent. BDI agents show a deliberative behavior and they are able to plan and pursue multiple objectives in parallel.

3.3 Utility-based Decisions

A i-th trip request tr_i contains the following information:

$$tr_i = (id_i, type_i, VATime_i, l_{start_i}, l_{end_i}) \quad (1)$$

It comprises a trip ID id_i, the type of trip $type_i$ set to "$CUSTOMER\_TRIP$", the desired vehicle arrival time $VATime_i$ to pickup the customer, a start location l_{start_i}, and an end location l_{end_i} of the trip. The vehicle agent aims to commit itself only to tr_i's that seem beneficial. A j-th vehicle agent va_j is defined as

$$va_j = (id_j, l_j, battery_j) \quad (2)$$

containing an id_j, a current geolocation l_j and a battery value $battery_j$. It has different options for a trip request tr_i that is advertised for bids or that is pre-assigned to va_j. The utility function for a trip request $u(tr_i)$ balances three relevant criteria, the journey to the customer u_{dist}, the battery power status $u_{battery}$ and the trip history u_{pts} which are weighted by w_1, w_2 and w_3. It evaluates each option to deal with tr_i in the agent's current situation as follows:

$$\begin{aligned} u(tr_i) = w_1 * u_{dist}(tr_i) + \\ w_2 * u_{bat}(tr_i) + w_3 * u_{pts}(tr_i) \end{aligned} \quad (3)$$

The calculation of the first component is based on a distance measure d for geolocations. A *geolocation* l is defined as $l = (longitude, latitude)$ in decimal degrees (DeMers, 2008). The euclidean distance approximates the distance between decimal degrees with a 1 meter variation in every 2,500 meters distance (cmp. the discussion in (Erduran et al., 2019)). The agent measures the euclidean distance d from its current location $va_j.l$ to the location to pickup the customer $tr_i.l_{start}$. The utility u_{dist} normalizes the distance value by means of a bounding box around the entire territory of the MAS in order to achieve values between 0 and 1. *dmax* denotes the maximum possible distance between two points at the borders of the territory:

$$u_{dist}(tr_i) = \frac{dmax - d(va_j.l, tr_i.l_{start})}{dmax} \quad (4)$$

As the second component, $u_{battery}$ considers the battery consumption of a potential trip in a rough approximation. Assuming a linear decrease of battery during traveling, the battery consumption in terms of number of charge units is directly derived from the travel time. The *travel time* between two geolocations l_x, l_y at a constant velocity v is estimated as:

$$travel\_time(l_x, l_y) = \frac{d(l_x, l_y)}{v} \quad (5)$$

The agent calculates the travel time for a potential round trip ($tr_i.type = round$) summing up the time for the journey to the customer $d(va_j.l, tr_i.l_{start})$, the trip itself $d(tr_i.l_{start}, tr_i.l_{end})$ and the journey back to the initial position $d(tr_i.l_{end}, va_j.l)$.

The utility measures the decrease of the current battery level $va_j.battery$ by the battery consumption for tr_i under consideration of $bpc_{i,j}$ which is the battery power consumption of tr_i processed by va_j. It is derived by the sum of $d(va_j.l, tr_i.l_{start})$ and $d(l_{start}, l_{end})$. We assume that a full battery contains 100% of power neglecting specific energy units. Since $bpc_{i,j}$ is proportional to the distance driven, we consider this as a percentage value reflecting the battery consumption related to the full battery. In case the current battery level $va_j.battery$ is too low to fulfill the trip, the utility takes the value $-\infty$. Since we consider a constant velocity, the battery consumption is derived by the distance driven and the time needed. In $u_{battery}$ the battery consumption is multiplied with a weight B_{factor} resulting to a utility score. We consider a distinction concerning the battery power level $va_j.battery$ of the vehicles in 3 levels:

$$B_{factor} = \begin{cases} 1.0, & va_j.battery > 80\% \\ 0.75, & 80\% \geq va_j.battery \geq 30\% \\ 0.1, & va_j.battery < 30\% \end{cases} \quad (6)$$

The B_{factor} is set to rate battery lifetime friendly thresholds higher. These thresholds are also used in other works (Ahadi et al., 2021; Zhang et al., 2020; Dlugosch et al., 2020). A $va_j.battery$ beyond the threshold gets a higher score to create incentives for the trip to reach the threshold. The function for the battery utility is defined as follows:

$$u_{bat}(tr_i) = \begin{cases} -\infty, & va_j.battery < bpc_{i,j} \\ B_{factor} * (\frac{bpc_{max} - bpc_{i,j}}{bpc_{max}}), & else \end{cases} \quad (7)$$

For bpc_{max} a vehicle agent will consider the minimum of his own current capacity $va_j.battery$ and the battery consumption for the maximum possible trip length.

For the third component u_{pts}, we consider the punctuality of the vehicle agent arriving to the trip starting position. It is divided into 4 levels. Here we consider θ as a threshold in minutes the customer is ready to wait until the trip is canceled and η is a buffer time where the vehicle should arrive in this. The punctuality is the result of the difference between the vehicle arrival time desired by the customer $VATime_i$ and the estimated time of departure $etd(tr_i.l_{start})$, which is calculated before driving to the trip request. Thus, we consider the following distinctions:

$$u_{pts}(tr_i) = \begin{cases} 0.0, & etd(tr_i.l_{start}) + \theta > tr_i.VATime \\ 0.2, & etd(tr_i.l_{start}) - \theta < tr_i.VATime < etd(tr_i.l_{start}) \\ 0.6, & 0 < tr_i.VATime - etd(tr_i.l_{start}) < \eta \\ 1.0, & tr_i.VATime - etd(tr_i.l_{start}) > \eta \end{cases} \quad (8)$$

The punctuality utility is 0.0, when the vehicle would not be on time to the customer even with the customer waiting which is defined by θ. It is 0.2 when the vehicle has a slight delay, 0.6 when there is a small buffer time defined by the threshold η and 1.0 when there is enough buffer time.

4 EXPERIMENTS AND EVALUATION OF TRIP ASSIGNMENTS

The feasibility of the proposed agent model is investigated by some experiments with sample data. The experimental setting comprises 10 sample data sets derived from historical trip data as described in Subsection 4.3. The benefits of the trip assignment method in a MAS (cmp. Section 3) is investigated by comparing its results with those of a centralized trip assignment method as a baseline. We denote the utility-based negotiation method for trip assignment *neg*. The baseline method denoted by *greedy* just assigns each incoming trip request to the agent that is recently in the closest proximity to the customer.

4.1 Evaluation Criteria

To evaluate our hypothesis, we will consider the customer satisfaction and the battery consumption. Two measures are defined as evaluation criteria for the agent behavior. The *order dropout rate ODR* measures the rate of trip requests that have been dropped. This is a measure for the customer satisfaction. The *average travel distance ATD* measures the average travel distance to serve a trip. Thus, ATD is a measure for battery consumption (cmp. the discussion on the linear dependency between distance and battery in Section 3.3). Both measures are based on the notion of a single agent's event trace σ. An *event trace* $\sigma(va_j) =< e_0, e_1, ..., e_k >$ records a sequence of events where a single event e comprises an event type $e\_type \in \{START, PICKUP, DROP, PASS\_BY, REFILL\}$, a geolocation l the agent has visited, an arrival time ta and a departure time td for l. For example, an event trace of an agent va_j with an initial geolocation l_0 where the agent starts to visit further k geolocations is denoted as follows:

$$\sigma(va_j) = \left\langle \begin{pmatrix} e\_type_0 \\ ta_0 \\ l_0 \\ td_0 \end{pmatrix}, \begin{pmatrix} e\_type_1 \\ ta_1 \\ l_1 \\ td_1 \end{pmatrix}, ..., \begin{pmatrix} e\_type_k \\ ta_k \\ l_k \\ td_k \end{pmatrix} \right\rangle \quad (9)$$

The *individual order dropout rate odr* is calculated counting the number of trip requests an agent va_j has missed divided by the number of trip requests it has committed to fulfill for the time interval covered by event trace σ.

$$odr(\sigma(va_j)) = \frac{\text{\# Dropped trips}}{\text{\# All committed trips}} \quad (10)$$

The order dropout rate of the entire MAS *ODR* is the average of the individual *odr*'s of its agents. Please note that trips which have been delegated to another agent by negotiation are not counted by $odr(\sigma(va_j))$ since they are under the responsibility of the delegate.

The second measure uses the *individual average travel distance atd* an agent is driven per successfully served trip according to the event trace σ. The *overall travel distance otd* calculates the sum of the partial routes from geolocation to geolocation:

$$otd(\sigma(va_j)) = \sum_{i=0}^{k-1} d(l_i, l_{i+1}) \quad (11)$$

atd normalizes the *otd* of an agent by dividing it through the number of served trips.

$$atd(\sigma(va_j)) = \frac{otd(\sigma(va_j))}{\#Servedtrips} \quad (12)$$

Analog to *ODR*, the *ADT* of the entire MAS is formed by the average of the individual *atd* values.

4.2 Simulating the Event Trace

It would be very expensive to assess a committed assignment in situ, i.e. within a real cyber-physical system. Instead, we simulate trip requests for the evaluation of the trip assignment process. Every agent records its committed trip requests in a trip plan. A trip plan w includes trip requests to serve a customer request as well as refill trips to the charging station. Therefore we denote:

$$w = (tr_1, tr_2, ..., tr_m) \quad (13)$$

where $tr_i.tr_{type}$ is set to $CUSTOMER\_TRIP$ in case tr_i refers to a trip request from a customer and $REFILL\_TRIP$ if the trip request comes from the agent itself to recharge battery. Note, that the end location is empty for trips of type $REFILL\_TRIP$. From the simulated trip plan, an event trace can be computed. It is built from processing w sequentially to create the events for each tr_i. Each event records the simulated time of arrival eta as ta and the simulated departure time etd as departure time td. As a rule of thumb, trip requests are considered missed if the simulated time of arrival has a delay above a threshold θ. A more detailed description of the build process of the simulated event trace $\sigma_e(va_i)$ can be found in the appendix as an algorithm in pseudo code (Alg. 1).

4.3 Experimental Trip Data

Trip requests from customers arise on a specific time and place in the considered area. In our ride-hailing experiment, the trips start and end at the University campus having a certain length with origin and destination coordinates. We create 10 samples containing the customer trip requests, where the largest sample contains 318 trips and the smallest 178 trips. The coordinates of the trips as well as the samples are generated from a station based bike sharing data set[1]. The data set is prepared and generated with random coordinates, to realize a free floating scheme, based on a method which is described in (Erduran et al., 2019). To achieve a higher frequency of requests, the real data from a week have been merged to a one-day data sample. A single trip tr in a sample is described as introduced in Section 3.3.

4.4 Experimental Prototype and Results

The MAS is implemented with *JADE* (Bellifemine et al., 2008). We set up the experiments for comparing agent communication according to the following structure: Each data sample described in 4.3 is used in 4 different configurations leading to 40 runs in total. We use two different amounts of participating vehicle agents 3 and 7 and both with and without negotiation denoted by neg_3, $greedy_3$, neg_7, and $greedy_7$. In all configurations, the trips are initially assigned by the

[1] https://data.deutschebahn.com/dataset/data-call-a-bike, last access: Sept 19, 2021.

area agent to the vehicle agent that is located closest to the customer. In the configuration where the agents are allowed to negotiate the trikes will use the utility function (cmp. Section 3.3) to decide if they will commit the trip by themselves or start the contract net protocol to delegate them to another vehicle agent. In the case where negotiation is not allowed every vehicle agent is forced to commit the trips as they were assigned to them by the area agent. In every setup the vehicle agents drive with a velocity of v = 3.6 km/h. For the evaluation, we use a θ of 4 minutes to decide if a vehicle agent arrives in time on the requested location. For each run, the vehicle agents start with a fully charged battery. The battery level is considered by the utility function. The refill trips, however, have not yet been implemented in our experimental prototype. The experiments are processed with a desktop computer, which contains an Intel Core i5-9500 and 32 GB DDR4-RAM. The source code can be found on GitHub[2].

The results for the considered ten samples are summarized in Table 1, where the ATD, ODR as well as the amount of lost and committed trips for every configuration is shown. In Fig. 1, only in one sample (VIII) the greedy approach leads to less lost trips than the approach with negotiation. In the 9 other sample runs, using negotiation leads to less lost trips than the greedy approach. In Fig. 2, the negotiating approach has in eight samples better results than the greedy. In the samples VI & VII neither of the two approaches cause lost trips. The overall results for 7 vehicle agents are better than for 3. These results are not only evident when looking at the summary. As expected, a higher amount of participating vehicle agents decreases the amount of lost trips. The more agents are located in the environment, the higher is the chance that there is an unused vehicle when a new trip request occurs. Comparing the results of 3 versus 7 vehicle agents, there is a small improvement of the ATD with 3 agents but a decline with 7 agents. This is probably due to the fact that 7 agents cause a very densely populated operational area. Since the amount of committed trips in both cases are the same, the ODR for among all samples is significantly smaller even in cases without negotiation. Obviously, more available vehicle agents reduce the cases in which the customer can not be reached in time. However, a small reduction of ODR can be observed, when negotiation is considered. Furthermore, we presume that negotiation is worthwhile with less amount of vehicles since the ODR decreases more for 3 vehicles. The interpretation of the results of our experiments requires a thorough analysis of the considered utility function of the vehicle agent. Concerning the equal weighting of the three components with $\frac{1}{3}$, we used this as a preliminary value. Each component of the utility function, which is explained in Section 3.3, incentives the vehicle agent towards a rational behavior. The distance utility u_{dist} reflects the situation that the closer a vehicle is to the start of the trip request, the higher is the utility score. Next, the battery utility u_{bat} impedes the vehicle to run out of battery and incentives the more balanced usage of all participating vehicles as well as staying in a healthy battery power range. For the last component u_{pts}, the behavioral intention is to cross out upcoming trips, where the vehicle arrival delay is bigger than the assumed time, the customers would wait. Those trips are also delegated to other vehicles with the used utility function.

To sum up, these three utility components reflect the leaning rational behavior of the vehicles. Although it can be expanded with more components leading to a more informed behavior, an arising disadvantage could be complex interpretation of such results. As a preliminary approach, we therefore set a basic utility configuration since our next step would be integrating a learned behaviour instead of the utility function. Concerning the scalability, a more complex simulation with more trip request samples and a larger operational area is required.

Table 1: Simulation results containing the sum of all 10 trip request samples.

config	ATD	ODR	trips lost	# trips
neg_3	**451.066**	**1.57%**	**38**	2416
$greedy_3$	462.064	4.55%	110	2416
neg_7	453.043	**0.08%**	**2**	2416
$greedy_7$	**447.881**	1.03%	25	2416

5 FUTURE WORK AND CONCLUSION

In this paper, we have presented a MAS of autonomous vehicles for managing and negotiating trip assignments in a ride-hailing scenario. An agent model has been introduced where each vehicle is represented by a BDI agent. In the 'think' phase, the agent makes a decision whether to negotiate on incoming trip requests based on a utility function, balancing the customer satisfaction concerning the pickup time and battery consumption. The negotiation follows the contract net protocol and its process is evaluated by simulating the agent behavior for experimental data created from real historical trips. The experimental results provide a proof of concept for the application of MAS in a novel MaaS system offering

[2]https://github.com/WI-user/ICAART22-submission

Algorithm 1: Pseudocode for the estimated event trace $\sigma_e(va_j)$.

Data: $w = (tr_1, tr_2, ..., tr_m)$; // a list of scheduled trip requests
t_0 ; // the time when starting to drive
l_0 ; // location of agent va_j at time t_0
θ ; // the threshold for max. tolerable delay
$REFILL\_DURATION$; // the average duration of recharge
Result: $\sigma_e(va_j)$; // a sequence of estimated events $< e_0, e_1, ..., e_k >$

$e_0 := (START, t_0, l_0, t_0)$ $\sigma := < e_0 >$ $n := 0$

for $g \leftarrow 1$ **to** m **do**
  $e\_type := tr_i.type$
  switch e_type **do**
    case $CUSTOMER\_TRIP$ **do**
      n++ $l := tr_i.l_{start}$ $eta := etd_{n-1} + travel\_time(l_{n-1}, l)$ **if** $eta \leq tr_i.VATime + \theta$ **then**
        if $eta < tr_i.VATime$ **then**
          $etd := tr_i.VATime$; // arrived too early
        else
          $etd := eta$
        end
        $e_n := (PICKUP, eta, l, etd)$ $\sigma.append(e_n)$; // e_n derived from start of tr_i
        n++ $l_{end} := tr_i.l_{end}$ $eta := etd + travel\_time(l, l_{end})$ $e_n := (DROP, eta, l_{end}, eta)$ $\sigma.append(e_n)$;
        // e_n derived from end of tr_i
      else
        $e_n := (PASS\_BY, eta, l, eta)$ $\sigma.append(e_n)$; // having missed the customer
      end
    end
    case $REFILL\_TRIP$ **do**
      n++ $l := tr_i.l_{start}$ $eta := etd_{n-1} + travel\_time(l_{n-1}, l)$ $etd := eta + REFILL\_DURATION$ $e_n :=$ $(REFILL, eta, l, etd)$ $\sigma.append(e_n)$ // charging completed
    end
  end
end

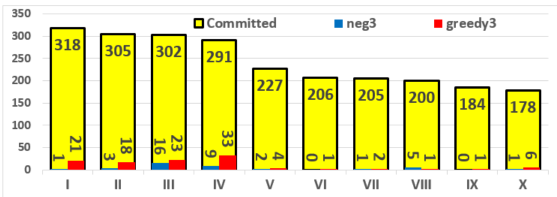

Figure 1: Amount of lost trips with 3 processing vehicle agents.

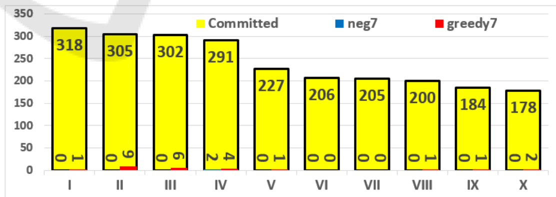

Figure 2: Amount of lost trips with 7 processing vehicle agents.

trip services for customers. Concerning our hypothesis, it can be said that the *ODR* in the negotiation approach is less than in the greedy approach. Since the energy consumption is derived from the *ATD*, where it is only less in the case with 7 agents, we can state that the experiments partially confirmed our hypothesis. A plausible reason for this circumstance could be the possibility that the negotiation approach processes trips with longer distances, which would be dropped out in the greedy approach and therefore leading to an decrease of its *ATD*. Our goal for the future is to use our vehicle agents to control a small fleet of real autonomous E-Trikes to prove our theory in real world. We want to extend our agent architecture and add components to make a learning based BDI agent. During this process, we will add an intermediate layer to interact with a full simulation platform like AMoDeus (Ruch et al., 2018). A complex simulation will also allow the comparison of our algorithms for the agent behaviour and the utility function from other works. The contribution of our work highlights the feasibility of an agent-oriented approach for ride-hailing stipulates multiple lines for future research on the distributed development of MAS.

REFERENCES

Ahadi, R., Ketter, W., Collins, J., and Daina, N. (2021). Siting and Sizing of Charging Infrastructure for Shared Autonomous Electric Fleets. *AAMAS*.

Bazzan, A. L. C. and Klügl, F. (2014). A review on agent-based technology for traffic and transportation. *The Knowledge Engineering Review*, 29(3):375–403.

Bellifemine, F., Caire, G., and Greenwood, D. (2008). *Developing multi-agent systems with JADE*. Wiley series in agent technology, Chichester, reprint. edition.

Certicky, M., Jakob, M., Pibil, R., and Moler, Z. (2014). Agent-based Simulation Testbed for On-demand Mobility Services. *Procedia Computer Science*, 32:808–815.

Danassis, P., Filos-Ratsikas, A., and Faltings, B. (2019). Anytime Heuristic for Weighted Matching Through Altruism-Inspired Behavior. *IJCAI 2019*, pages 215–222.

Deljoo, A. (2017). What Is Going On: Utility-Based Plan Selection in BDI Agents. *The AAAI-17 Workshop on Knowledge-Based Techniques for Problem Solving and Reasoning*.

DeMers, M. N. (2008). *Fundamentals of geographic information systems*. John Wiley & Sons.

Dlugosch, O., Brandt, T., and Neumann, D. (2020). Combining analytics and simulation methods to assess the impact of shared, autonomous electric vehicles on sustainable urban mobility. *Information & Management*, page 103285.

Erduran, O. I., Minor, M., Hedrich, L., Tarraf, A., Ruehl, F., and Schroth, H. (2019). Multi-agent Learning for Energy-Aware Placement of Autonomous Vehicles. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pages 1671–1678, Boca Raton, FL, USA. IEEE.

Fu, K., Meng, F., Ye, J., and Wang, Z. (2020). CompactETA: A Fast Inference System for Travel Time Prediction. In *ACM SIGKDD 2020*, pages 3337–3345, Virtual Event CA USA. ACM.

Jaroslaw Kozlak, S. P. and Zabinska, M. (2013). Multi-agent models for transportation problems with different strategies of environment information propagation. *PAAMS 2013, Springer Berlin Heidelberg*.

Kamargianni, M., Li, W., Matyas, M., and Schäfer, A. (2016). A Critical Review of New Mobility Services for Urban Transport. *Transportation Research Procedia*, 14:3294–3303.

Kowalski, R. and Sadri, F. (1996). Towards a unified agent architecture that combines rationality with reactivity. *Logic in Databases*, 1154.

Kowalski, R. and Sadri, F. (1999). From Logic Programming towards Multi-agent systems. *Annals of Mathematics and Artificial Intelligence*, 25(3/4):391–419.

Malas, A., Falou, S. E., and Falou, M. E. (2016). Solving On-Demand Transport Problem through Negotiation. *Proceedings of the Summer Computer Simulation Conference*, page 7.

Pavone, M. (2015). Autonomous Mobility-on-Demand Systems for Future Urban Mobility. In Maurer, M., Gerdes, J. C., Lenz, B., and Winner, H., editors, *Autonomes Fahren*, pages 399–416. Springer Berlin Heidelberg.

Qin, Z. T., Tang, X., Jiao, Y., Zhang, F., Xu, Z., Zhu, H., and Ye, J. (2020). Ride-Hailing Order Dispatching at DiDi via Reinforcement Learning. *INFORMS Journal on Applied Analytics*, 50(5):272–286.

Rao, A. S. and Georgeff, M. P. (1995). BDI Agents: From Theory to Practice. *ICMAS*.

Rüb, I. and Dunin-Keplicz, B. (2020). Basta: Bdi-based architecture of simulated traffic agents. *Journal of Information and Telecommunication*, 4(4):440–460.

Ruch, C., Horl, S., and Frazzoli, E. (2018). AMoDeus, a Simulation-Based Testbed for Autonomous Mobility-on-Demand Systems. In *2018 ITSC*, pages 3639–3644, Maui, HI. IEEE.

Silva, L. d., Meneguzzi, F., and Logan, B. (2020). BDI Agent Architectures: A Survey. In *IJCAI 2020*, pages 4914–4921, Yokohama, Japan.

Smith, R. G. (1980). Communication and Control in a Distributed Problem Solver. *IEEE Transactions On Computers*.

Yu, M. and Zhang, Y. (2010). Multi-agent-based Fuzzy Dispatching for Trucks at Container Terminal. *International Journal of Intelligent Systems and Applications*, 2(2):41–47.

Zhang, H., Sheppard, C. J., Lipman, T. E., Zeng, T., and Moura, S. J. (2020). Charging infrastructure demands of shared-use autonomous electric vehicles in urban areas. *Transportation Research Part D: Transport and Environment*, 78.