Transportation Research Interdisciplinary Perspectives 7 (2020) 100190

Contents lists available at ScienceDirect

Transportation Research Interdisciplinary Perspectives

journal homepage: https://www.journals.elsevier.com/transportation-research-interdisciplinary-perspectives

Comparison of competing market mechanisms with reinforcement learning in a carpooling scenario

Thomas Pitz, Deniz Kayar *, Wolf Gardian, Jörn Sickmann, Hasan Alkaş

Rhine-Waal University, Wiesenstraße 35, D-47533 Kleve, Germany

ARTICLE INFO

Article history:
Received 5 March 2020
Received in revised form 14 July 2020
Accepted 30 July 2020
Available online xxxx

Keywords:
Multi-agent simulation
Reinforcement learning
Traffic
Carpooling
Auction
Market

ABSTRACT

In this paper a multi-agent simulation was implemented to analyze the dynamics of different market mechanisms with a Reinforcement Learning algorithm in the context of a carpooling market. The agents in the simulation, car owners (COs) and non car owners (NCOs), had to sell or buy a car seat for multiple rounds by picking one of two possible mechanisms: Dutch Auction or Fixed Price. In the beginning of the simulation the agents have no information about the efficiency of these mechanisms and they are chosen with the same probability. In the course of the simulation a Reinforcement Learning algorithm alters the agents' preferences for the two mechanisms depending on their cumulative payoffs. The key finding is that sellers have a clear preference for the Dutch auction mechanism with differing degrees dependent on the seller/buyer ratio. Buyers on the other hand have no significant preference for any mechanism. If these results are replicable, they suggest that an increased utilization of the Dutch auction could lead to an expansion of the carpooling market, increasing its impact as an alternative means of transportation.

1. Introduction

Carpooling, also known as ridesharing, is the term used to describe a pool of at least two people who travel together to a common destination using a vehicle owned by one of them (Teal (1987)). Carpooling (as part of the Sharing Economy) is a sustainable concept to efficiently cover distances cooperatively. From a societal viewpoint, it is associated with numerous benefits such as reduced energy consumption and Greenhouse Gas (GHG) emissions, improved air quality as well as reduced road and parking infrastructure demand.

Economically speaking, carpooling markets are so-called two-sided markets. The theory of two-sided markets has been developed to analyze markets in which two distinct user groups (sets of economic agents) interact through an intermediary platform and indirect network effects are important between both user groups (see Rochet and Tirole (2003) and Rochet and Tirole (2006) for pioneering work). While in traditional one-sided transport markets firms compete only for the customers and hire workers (e.g. taxi cab, train or bus drivers), in the carpooling market platform providing firms compete both for passengers and drivers (Cohen and Zhang (2017)) and the core business of the platform is to bring both sides “on board”. The success of a carpooling network depends particularly on the number of active users on both sides of the market equation. Therefore a carpooling provider must implement an efficient allocation mechanism to attract drivers as well as passengers (Arning et al. (2013), Knapen et al. (2013) and Galland et al. (2014)). This describes a key research focus of the formal economic two-sided markets literature, which in particular analyzes pricing choices. Pricing in two-sided markets is more complex than in one-sided markets and is strongly affected by indirect network effects (Rysman (2009)). One of the most prominent allocation mechanisms for online carpooling platforms works as follows: Drivers can offer free seats for a specific route and search for potential passengers who can accept offers or request free seats. The prices to be paid by the passengers can be set by the car owners (who offer available seats) only within certain limits.

The main question investigated in the present paper is: Are there more efficient allocation mechanisms for carpooling platforms? For this purpose we implemented a multi agent framework for simulating various types of competing market mechanisms (allocation mechanisms) to allocate drivers and passengers. In the presented study we compared two market mechanisms. The first mechanism we refer to as Fixed Price is adopted in practice by the largest carpooling platform Blablacar (Farajallah et al. (2016) and Shaheen et al. (2017)) and can be viewed as a standard mechanism. The second mechanism is a Dutch Auction. A Dutch auction has the property that it ends immediately after a bidder proposes a price and could thereby reduce waiting times for the passenger. A similar approach to the one presented in this paper can be seen in Hailu and Schilizzi (2004), where they investigated whether an auction performs better over time then a fixed-price scheme. In the following subsections we discuss the operationalized Reinforcement Learning algorithm, which

* Corresponding author.
E-mail address: deniz.kayar@hochschule-rhein-waal.de (D. Kayar).

http://dx.doi.org/10.1016/j.trip.2020.100190
2590-1982/© 2020 The Author(s). Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

is used for agent decision making between the two allocation mechanisms in the simulation.

1.1. Learning algorithm: reinforcement learning

The agents in this paper can choose between the two mechanisms, Dutch Auction and Fixed Price. They can be seen as the agents' two strategies, which have to be selected in each period of the simulation. After each period the outcome of a strategy is used to update the propensities in the Reinforcement Learning algorithm. That means future decisions depend on preceding outcomes. This algorithm is described in detail in Section 2.

The agents' decision behavior was altered by a Reinforcement Learning process (Bush and Mosteller (1955)). The Reinforcement Learning algorithm was chosen in this paper because except from the initialization it is one of the few basic parameter free learning algorithms. Additionally in (Chmura (2005), p.107), Chmura and Pitz (2007), Selten et al. (2006), Chmura et al. (2012) and Goldbach et al. (2019) it was shown that it is an appropriate algorithm to describe human behavior, where participants in experiments are confronted with a scenario of interactive and iterative binary decision making. For example observed fluctuations around a Nash equilibrium could be predicted quite well. The software presented in this paper has an open architecture, which allows implementing further types of learning algorithms.

In the Reinforcement Learning process, agents' attitudes are represented as natural numbers, which are called propensities. In each period propensities are altered proportionally to the payoffs of previous periods. The normalized propensities are used as probabilities, which control future strategic decisions. The system determines the probabilities for specific future decisions based on past experiences. There is a positive correlation between successful strategies and the probability of choosing the same strategy again (Moriarty and Schultz, 1999). This approach has gained significant attention and extensive study in the last 20 years (Kaelbling et al., 1996). Particularly in economic literature it was discussed in Cross (1973), Harley (1981), Arthur (1991), Arthur (1993), Roth and Erev (1995), Erev and Roth (1998), Camerer and Ho (1998), Camerer and Ho (1999) and Chen and Hsieh (2009).

There are different applications of Reinforcement Learning with the general aim to solve similar sequential decision tasks through non-supervised interactions with the environment ((Moriarty and Schultz, 1999, p. 242); (Barto et al., 1990). Reinforcement Learning is widely used in the field of Artificial Intelligence, especially in combination with Artificial Neural Networks (ANN) and Deep Learning. Besides Supervised and Unsupervised Learning, it is an additional paradigm to train an ANN.

1.2. Allocation mechanisms and related work

The Dutch Auction (having its origin in the 17th century flower sales of the Netherlands) is a sequential descending bid auction with a successively decreasing price. Aalsmeer, for example, the largest flower auction in the world, has an annual turnover of 4.6 trillion euros and 2956 employees. They sold 12.5 trillion flowers and plants in 2016 (FloraHolland, 2018). In a Dutch Auction the first bidder who accepts the price wins and the auctions terminates immediately. In the present paper Dutch Auctions on the simulated carpooling market will be conducted simultaneously (the quantity of auctions depends on the number car owners who decide to find a passenger via auction). Its advantage in comparison to an English Auction (ascending) is, that a buyer can stop the auction if the reservation price is reached. Therefore, a buyer is not forced to wait for the end of the auction to find out if a bid was successful. For this reason Dutch Auctions are more suitable for carpooling markets, where supply and demand exist only for limited time periods. The Dutch Auction is a sequential auction where bids are made in steps, so after every round additional information about bidding behavior will be disclosed.

There are already some approaches, which apply Reinforcement Learning to simulated bidding behavior in auctions. See for example Pentapalli (2008), Hailu and Schilizzi (2004) or Guerci et al. (2013, 2014) for the evaluation of the performance of auctions under dynamic settings using agent-based modeling. In contrast we focus not on bidding behavior but on the decision between two market mechanisms.

Despite the high efficiency of auctions, which is also demonstrated in the presented simulations, existing carpooling platforms generally do not use Dutch Auctions but rather Fixed Price (Posted prices) as an allocation mechanism. The reason could be that carpooling providers assess Dutch Auctions as less understandable and thus unsuitable for the average consumer. Nevertheless this decision needs further investigation.

The Fixed Price (price for a free seat) in the presented simulation corresponds to the individual reservation price of a car owner. A buyer can decide to take the lift for the specified price or not. Bargaining (a further alternative mechanism) is not possible with posted prices. Analysis of the use of auctions versus posted prices has a rather long tradition (for pioneering work see e.g. Wang (1993, 1998). Kultti (1999)(see also Kultti and Virrankoski (2004)), demonstrating an equivalence of auctions and posted prices under specific conditions (e.g. large market and homogeneous buyers), whereas compared to bargaining, posted -price markets dominate.

While auctions are associated with optimal results if transaction costs are zero, posted prices may be superior if auctions are costlier. Observations in the real world appear to support this theoretical insight. Whereas Fixed Price seems to be the dominating allocation mechanism for standardized goods with a commonly understood value, auctions are typically used for the sale of more specialized and (often) more scarce goods. Auctions have also traditionally been the predominant mechanism used for the sale of goods over the internet. Virtual internet platforms allow for a significant reduction of the transaction costs associated with auctions compared to physical platforms. However, more recently, there has been an increasing tendency towards the use of posted prices within online platforms (Einav et al., 2018).

1.3. Contributions

In the following sections we describe the dynamics of two complex, competing allocation mechanisms. There already exists a well-developed mathematical framework for equilibrium and revenue analysis for each mechanism (for example (Ricardo, 1817) for the Fixed Price Mechanism and (Klemperer, 1999) for Auctions). Nevertheless, the advantage of an "'experimental'" multi agent simulation is, that it focuses on aspects which can be only analyzed mathematically, taking enormous effort: An example of this is a dynamic combination of both mechanisms, their competing behavior over many periods and the effect of variation of a high number of parameters. The authors believe that this method – in addition to standard mathematical theories, provides a profound insight into the problem of designing market and allocation mechanisms.

2. Methodology and problem formulation

In the presented simulation a fixed number of agents have to travel from one location to another by car or train. This is repeated for a fixed number of periods T. The agents are divided into two groups: car owners (COs) and non car owners (NCOs). The first group are the drivers and sellers of seats and the second group are potential passengers and buyers of seats. In each period t, with $t \in [1,...,T]$, each CO offers one of his car seats to NCOs. Each agent, whether CO or NCO, has a reservation price p_r which is a natural number between $[p_{r_{min}}, p_{r_{max}}]$. For each agent in each period, p_r is independently and randomly chosen according to a uniform distribution on $[p_{r_{min}}, p_{r_{max}}]$.

In each period, each agent has to choose randomly between two strategies: Fixed Price or Dutch Auction. In the first period the mechanisms are picked with the same probability. In the following periods the probabilities change by updating propensities due to a Reinforcement Learning algorithm (Section 2.4). The propensities depend proportionally on the cumulative payoffs of the previous periods. The payoff is defined

in Section 2.3. COs who chose mechanism x in period t can only be matched with NCOs who also chose the mechanism x in period t.

If NCOs do not find a lift they have to travel by train. The train price is p_{train} with $p_{train} \geq p_{r_{max}}$.

2.1. Dutch auction

If a CO i chooses Dutch Auction in period t, the price of the auction for i's seat is set to a start price p_1 with $p_1 > p_{r_{max}}$ in the first step of this auction. The price of a Dutch Auction is decreased in each step s of the auction by the step width $w \in [w_{min}, w_{max}]$, a natural number which is determined at the beginning of a Dutch Auction and stays constant throughout the lifetime of the auction. If the price in step s is larger than the reservation price of all NCOs, who chose Dutch Auction, the next step $s + 1$ is started and the auction price lowered by w. The maximal step count for Dutch Auctions is set to s_{max} in order to ensure that each auction stops.

If the current price of the auction is smaller than or equal to the reservation price of a randomly chosen NCO j, who has chosen Dutch Auction, i's auction stops and j travels in i's car. Otherwise, if the current price in a given step s of the auction is smaller than i's reservation price the auction stops and no agent is found to travel in i's car.

The starting time is determined randomly for each auction with the parameter $\hat{s}$, which serves as a price increment, so that the auctions start asynchronously and the sequential character of carpooling auctions is simulated: Each Fixed Price or Dutch Auction starts at a predetermined start time $\hat{s} \in [\hat{s}_{min}, \hat{s}_{max}]$. Different price increments are used to represent different starting times.

It has to be noted, that bid behavior and risk attitude of NCOs in the present simulation is elemental. Also information changes during bidding phases do not alter the agents bid behavior or risk aversion. There are merely two lists - one for auctions and one for bidders. Auctions from the auction list are matched with bidders from the bidders list if the bidders reservation price is greater than or equal to the auctions current price. The algorithm starts with the top most auction and compares its price with the first bidders reservation price. If it doesn't fit, it checks the following bidders it finds one bidder that meets the requirement ((see Fig. 1) for example matchings).

2.2. Fixed price

For the Fixed Price mechanism seats are sold immediately without waiting for any auction mechanism (e.g. falling or rising prices) to take place. For a CO i in period t the price of i's seat is set to i's reservation price in that period. If there is a randomly determined NCO j, whose reservation price is larger or equal to the price of i's seat, j chooses i's offer. Otherwise, if the price of i's offered seat is greater than the reservation price of all NCOs, who chose Fixed Price, the Fixed Price offer is finished and no agent is found to travel in i's car.

At each time step the matching of an offer and a demand with fitting reservation prices is performed randomly. Technically the list of open offers and the list of potential buyers is mixed randomly in each step for both mechanisms.

2.3. Payoff

If the strategy chosen by CO i does not result in finding a NCO in period t, i has to travel alone and the seat could not be sold, i.e. the revenue for i in period t is 0. That means for a CO the total payoff is $0 - p_{r_i} = -p_{r_i}$.

If a NCO j does not find a seat in period t, j has to take the train and therefore has to pay the train price, which is higher than $p_{r_{max}}$. In conclusion the total payoff for j in period t is $(p_{r_j} - p_{train}) \leq 0$.

If otherwise an NCO j takes the seat in i's car for paying the price p in period t the resulting payoffs for these two agents in period t are: $\pi_i = p - p_{r_i}$ for i and $\pi_j = p_{r_j} - p$ for j.

2.4. Reinforcement learning

In the present paper the Reinforcement Learning described in Chmura and Pitz (2007) and Selten et al. (2006) was implemented, which can also handle negative payoffs. Each agent in the simulation chooses one strategy x out of n strategies for each period t of the simulation, where $t \in (1,..,T)$ and T denotes the number of periods. The probability P_x^{t+1}, that an agent chooses strategy x in period $t + 1$ is proportional to its propensity Q_x^{t+1}.

Whenever a strategy x has been chosen in period t the corresponding payoff π^t is added to the propensity Q_x^t if the payoff is positive. For positive payoffs the propensity can be interpreted as the payoff sum of the previous periods.

Initialization:

For each agent let $[Q_1^1, ..., Q_n^1]$ be the initial propensities, where n is the number of possible strategies.

Period 1:

Each agent chooses strategy x with probability

$$\frac{Q_x^1}{\sum_{y=1}^{n} Q_y^1} \tag{1}$$

where $Q_y^1 \varepsilon [Q_1^1, ..., Q_n^1]$, the initial propensities of all possible strategies.

Periods t + 1:

For each agent, let π^t be the payoff in period t and x the chosen strategy in period t. Then the propensities for period $t + 1$ are calculated as follows:

$$\text{CASE I}\,(\pi^t \geq 0) : Q_y^{t+1} := \begin{cases} Q_y^t + \pi^t, \text{if } y = x \\ Q_y^t, \text{if } y \neq x \end{cases}$$

$$\text{CASE II}\,(\pi^t < 0) : Q_y^{t+1} := \begin{cases} Q_y^t, \quad \text{if } y = x \\ Q_y^t + |\,\pi^t\,|, \text{if } y \neq x \end{cases}$$

Each agent chooses strategy x with probability

$$\frac{Q_x^{t+1}}{\sum_{y=1}^{n} Q_y^{t+1}} \tag{2}$$

In period 1 the probabilities for each strategy x are initialized with $1/n$ and propensities are initialized with 1. In the present paper $n = 2$ and the possible strategies are Dutch Auction and Fixed Price.

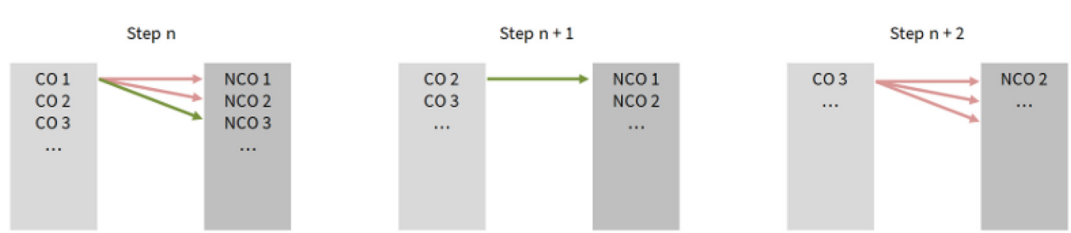

Fig. 1. Auction and bidder list matching (green arrows are successful matchings).

In Fig. 2 each step in a period of the simulation for a bidding agent is depicted. The algorithm is analogical for a selling agent. The algorithm is described in pseudocode in Algorithm 1. Other subroutines can be found in Section 6.

2.5. Implementation

The simulation has been implemented with Java. Single simulation runs are independent from each other, so the software has been designed multithreaded to accelerate the execution. That means that with increasing CPU core count more simulations can be executed at the same time. Therefore, depending on the CPU and the count of simulation runs the execution will vary in run time. The code is available online and can be accessed here: Kayar (n.d.). The software architecture is expandable by different distributions for the reservation prices, learning algorithms and allocation mechanisms. Right now there are uniform and normal distributions available for the generation of reservation prices, a reinforcement learning algorithm serves as the learning algorithm and Dutch Auction and Fixed Price are implemented as allocation mechanisms.

In Algorithm 1 the pseudo code for the simulation's main function is depicted. In the beginning the configuration file (input XML file) is read and all relevant variables are initialized with the values specified in the XML tags. For each period of a simulation the reservation prices and strategies are randomly determined. Depending on the chosen strategies all agents are put into one of four lists depending on whether they are NCOs or COs and whether their strategies are Dutch Auction or Fixed Price.

After these first initialization steps the "execAuctions" (Algorithm 2) function is called once for the two Dutch Auction lists and once for the two Fixed Price lists. The function is executed recursively. That means it calls itself (nested calls) until an exit condition is met. With each call the auctions and NCOs, that were matched in the superordinate function calls, are removed from the input lists.

Before the auctions are executed the input list sizes are checked. If one of them is empty it means either all COs found a buyer (count of COs is 0) or all NCOs have found a seat in a car (count of NCOs is 0) then the execution of "execAuctions" will be interrupted. For the first case (all seats are sold) the remaining NCOs take the train and are removed from the NCOs list. For the second case all remaining auctions are marked as finished, their foundBuyer flag is set to false and the auctions are removed from the COs list.

Algorithm 1
auctionSimulationMain().

```
Algorithm 1: auctionSimulationMain()
/* Declarations, read from input XML file                                  */
countPeriods ← 100 ;                                  // Number of periods
countCOs ← 25 ;                                  // Number of COs, e.g.  25 COs
countNCOs ← 100 ;                              // Number of NCOs, e.g.  100 NCOs
countAgents ← countCOs + countNCOs ;                // Total Number of agents
agentList ← NULL ;                             // List of agents(COs and NCOs)
minReserv ← 0 ;                                // Lower bound reservation price
maxReserv ← 100 ;                              // Upper bound reservation price
trainPrice ← 100 ;                                          // Train price
dutchAuctionCOs ← NULL ;                                // Dutch Auction COs
dutchAuctionNCOs ← NULL ;                              // Dutch Auction NCOs
fixedPriceCOs ← NULL ;                                   // Fixed Price COs
fixedPriceNCOs ← NULL ;                                 // Fixed Price NCOs
/* Initializations                                                         */
for i ← 0 to countAgents − 1 do
    agentList[i] ← newCO();
end
/* Execute simulation periods                                              */
for i ← 0 to countPeriods − 1 do
    for j ← 0 to countAgents − 1 do
        agent ← agentList[j];
        /* Determine new random reservation prices for agent               */
        agent.reservPrice ← rand(minReserv, maxReserv);
        if agent.reservPrice > trainPrice & agent.isNCO() then
            agent.reservPrice ← trainPrice;
        /* Get strategy for agent(Dutch or Fixed?)                         */
        agent.reinforcementLearning.getStrategy();
        /* Put agent in corresponding participants list                    */
        if agent.strategy == dutch & agent.isCO() then
            dutchAuctionCOs.add(agent);
        else if agent.strategy == dutch & agent.isNCO() then
            dutchAuctionNCOs.add(agent);
        else if agent.strategy == fixed & agent.isCO() then
            fixedPriceCOs.add(agent);
        else if agent.strategy == fixed & agent.isNCO() then
            fixedPriceNCOs.add(agent);
    end
    /* Execute all auctions for this period.  execAuctions() is
       recursive                                                           */
    execAuctions(dutchAuctionCOs, dutchAuctionNCOs, 0);
    execAuctions(fixedPriceCOs, fixedPriceNCOs, 0);
end
```

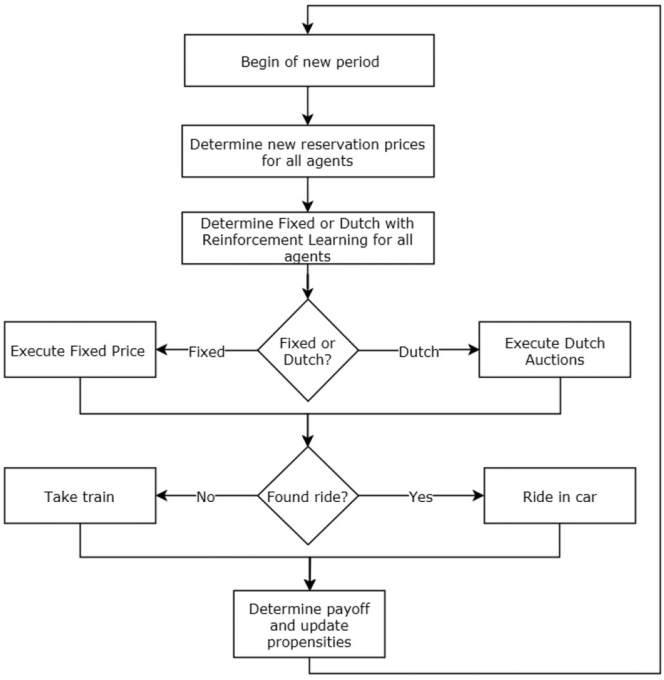

Fig. 2. One period of the simulation for a bidder.

If neither of the conditions are met all auctions that belong into the current global time step are randomly permuted and executed in the new order. The algorithm then again tries to match buyers with the currently available auctions.

The matching of buyers and auctions is described in Algorithm 3. First the current price of an auction is updated according to its step width. If the current price is lower than the reservation price of the owner, the auction is finished and no buyer was found for this auction. The payoff for the CO is 0 then and the auction is removed from the auction list. If the current price is still higher than the reservation price, the list of possible buyers is randomly shuffled and NCOs try to match with the current auction, starting with the first NCO in the list. If there is a match the payoffs, propensities and probabilities are updated for the involved COs and NCOs according to the Reinforcement Learning algorithm (Section 2.4).

Propensities are updated in Algorithm 4. The sum of payoffs for a strategy is increased by the current payoff if the strategy has been chosen for the current round. The cumulative sums over all strategies are updated accordingly.

Depending on the propensities the probabilities are updated in Algorithm 5. The CDF (cumulative distribution function) object is a map data structure, that holds the probability ranges for each strategy. In the first simulation step these ranges are [0,0.5] and]0.5,1] for the two possible strategies. In further rounds these ranges shift depending on the payoffs and propensities of chosen strategies and the evaluation of Formula 2.

Finally, Algorithm 6 generates a random number and chooses a strategy if the random number falls into its probability range.

Table 1
Simulation parameters.

Parameter	Description	Value
$\#COs$	Count of car owners	25
$\#NCOs$	Count of non car owners	100
$p_{r_{min}}$	Minimal reservation price	1
$p_{r_{max}}$	Maximal reservation price	100
p_{train}	Train price	100
p_s	Dutch auction start price	110
w_{min}	Minimal Dutch Auction price step width	1
w_{max}	Maximal Dutch Auction price step width	5
$\hat{s}_{min}$	Minimal Dutch Auction start time	1
$\hat{s}_{max}$	Maximal Dutch Auction start time	10

3. Results

Several simulations have been conducted: multiple runs with varying agent counts. All analyzes were done with R 3.3.2 (R Development Core Team, 2008). The data visualization library ggplot2 2.2.1 (Wickham, 2009) was used to generate the figures.

If not stated otherwise all simulation runs were executed with the parameters listed in Table 1. The train price was set to $p_{r_{max}} = 100$. Therefore, taking the train results in the lowest possible payoff for an NCO. In each period the reservation price for each agent was determined by a random uniform distribution between 1 and 100. The number of NCOs was set to 100 and the number of COs was set to 25. Dutch auction step widths were randomly chosen between 1 and 5 and starting times were randomly chosen between 1 and 10.

3.1. Multiple runs

The simulations was repeated 100 times independently and the mean values for payoffs and strategic preferences have been analyzed. As can be seen in Fig. 3 (left) there is a clear increasing preference for Dutch Auctions for COs in the course of the simulation runs.

The mean payoff ($\bar{\pi}_i$) for all COs (Fig. 3 (right)) is 12.85 with a mean inner standard deviation of 50.58 per period and simulation and a standard deviation of 9.96 over 1000 periods. These values are close to the values of the previous single simulation run. The mean payoff seems to stay almost constant throughout the simulation although there is a clear preference change to Dutch Auction early in the simulation runs. We assume that due to the cumulative behavior of Reinforcement Learning there is a noticeable effect on the preference but not on the mean payoff. The situation is not completely clear and this may be attributable to modeling limitations. Potentially it is possible to clarify this through deeper analysis of the model by variation of the Reinforcement Learning propensities.

For NCOs on the other hand the payoff starts negative and decreases in the first few periods of the simulation (Fig. 4 (right)). The average payoff for NCOs is -44.38 with a mean inner standard deviation of 35.27 per period and simulation and a standard deviation of 2.86 over 1000 periods.

NCOs stay indifferent regarding the Dutch Auction and Fixed Price over all periods and simulation runs (Fig. 4 (left)). On average 50.21 out of 100 NCOs chose Dutch Auction with a standard deviation of 5.17.

3.2. Multiple runs with varying car owner count

In this section a few sequences of multiple simulations with a different CO/NCO ratios are described. The ratios were varied as follows: There was a constant number of NCOs in each simulation: 100. The number of COs on the other hand took values between 10% up to 200% of the number of NCOs. For each of these 5 configurations we conducted 100 simulation runs with each run consisting of 500 periods, with the first period being indexed with 0 and the last period with 499 (Fig. 5). In Tables 2 and 3, 5 of the results of the last periods for the different configurations are listed .

In Fig. 6 (left) the fraction of COs, who chose Dutch Auction for all five configurations is depicted. In all scenarios the preferences converge

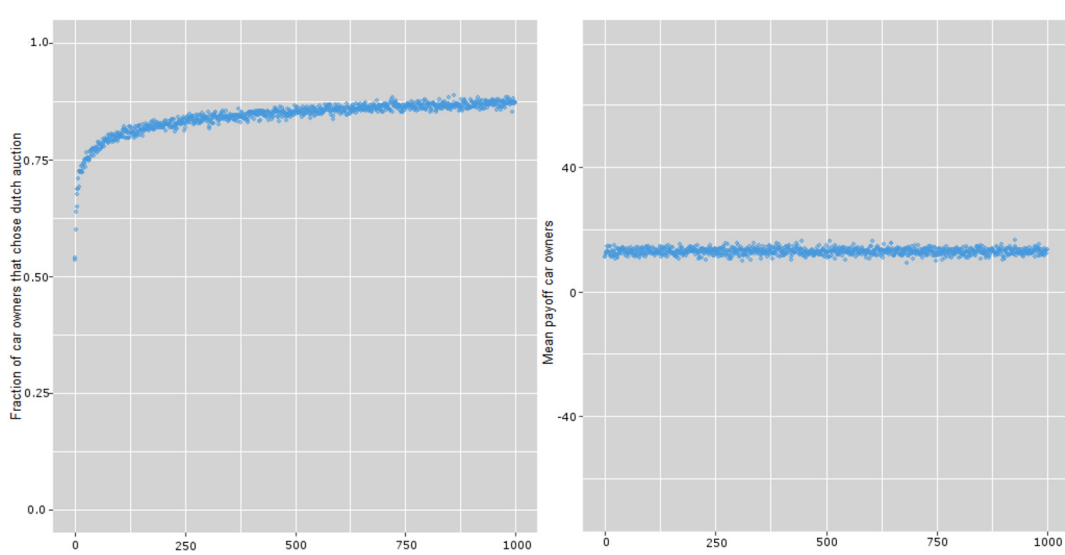

Fig. 3. Mean fraction of COs who chose Dutch Auction (left) and their mean payoff (right).

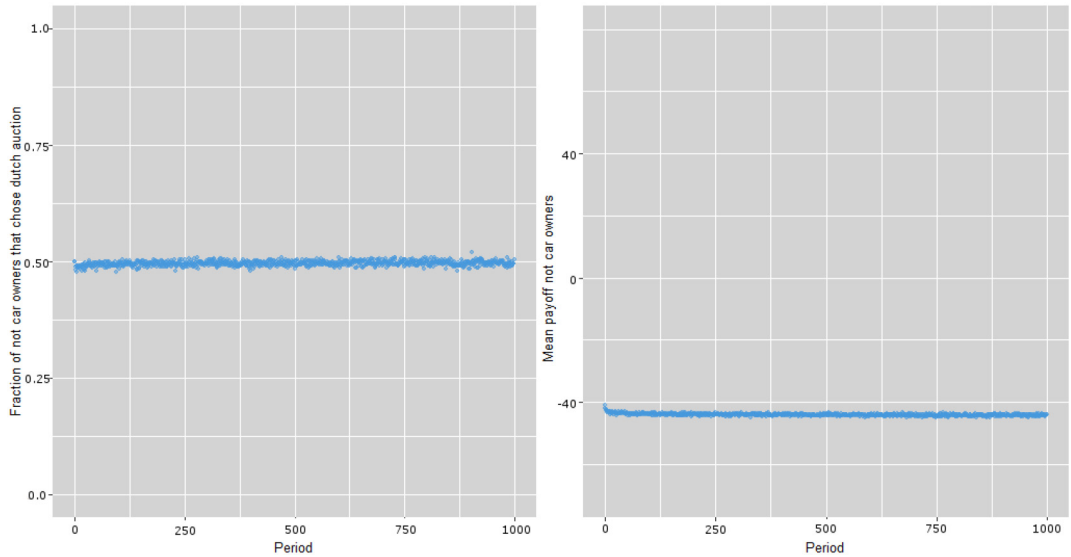

Fig. 4. Mean fraction of NCOs who chose Dutch Auctions (left) and their mean payoff (right).

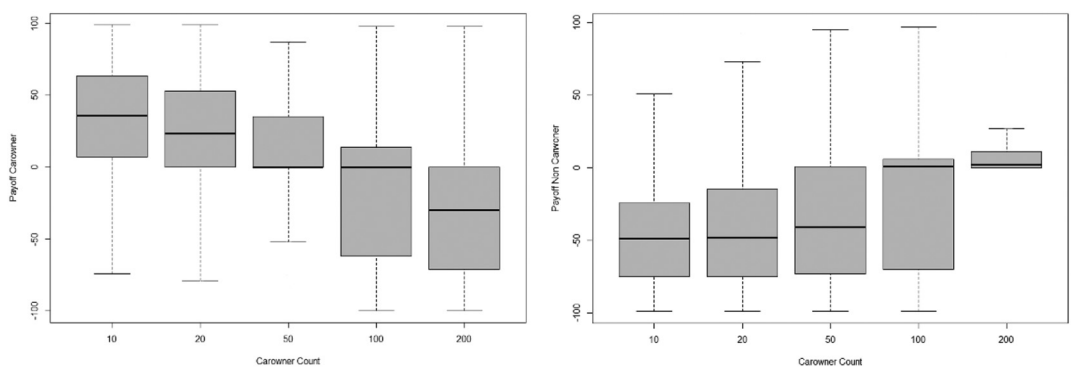

Fig. 5. The payoffs from Table 3 as boxplots.

towards the Dutch Auction in different degrees. The convergence seems to depend on the CO/NCO ratio. For simulations with high numbers of COs the preference is lower than the ones with small numbers of COs, but it still holds true (see also Fig. 7). This is in line with economic intuition in regard to supply and demand ratios.

In the simulations with high CO counts (100 and 200) there is one peak in the starting periods where the preference stops growing and falls back to a specific value, settles and stays there until the end of the simulations. In these case the Dutch Auctions seem to be overrated in the first few periods. In the other scenarios (10, 20 and 50) the probabilities for choosing Dutch Auction is continually increasing without any relapse.

Table 2
Results for final periods with varying CO counts (COs: Number of COs; NCOs: Number of NCOs; FoundBuyers: Mean number of COs, that found a buyer; TookTrain: Mean number of NCOs, that took the train; NCO Dutch: Mean fraction of NCOs that Dutch Auction; CO Dutch: Mean fraction of COs that chose Dutch Auction).

COs	NCOs	FoundBuyers	TookTrain	NCO Dutch	CO Dutch
10	100	9.08	90.92	0.49	0.97
20	100	17.14	82.86	0.49	0.89
50	100	39.36	60.64	0.49	0.71
100	100	67.57	32.43	0.47	0.59
200	100	89.59	10.41	0.45	0.55

In Fig. 6 (right) the payoffs for COs for the different configurations are depicted. As expected there is a negative correlation between CO numbers and mean payoff. With decreasing CO numbers the mean payoff increases. That means COs were less likely to sell a seat, which is caused by the higher number of competing sellers. With increasing numbers of COs less seats will be taken and the mean payoff for COs ($\overline{\pi}_i$, see Section 2.3) decreases. The observed higher variance in the mean payoff for low CO numbers is caused by the lower number of data points for calculating the mean.

Fig. 8 (left) shows the fraction of NCOs who chose Dutch Auctions in the different scenarios. With increasing numbers of COs, the NCOs show a preference for Fixed Price. Nevertheless, the probability of choosing Dutch Auction is lower for NCOs than for COs.

Table 3
Mean Payoffs for final periods with varying CO counts ($\overline{\pi}_i$: Mean payoff for COs; $\overline{\pi}_j$: Mean payoff for NCOs; σ: standarddeviation).

COs	NCOs	$\overline{\pi}_i(\sigma)$	$\overline{\pi}_j(\sigma)$
10	100	30.49 (16.26)	−48.09 (2.96)
20	100	18.14 (10.02)	−46.19 (2.89)
50	100	0.74 (6.09)	−35.48 (3.33)
100	100	−13.93 (3.68)	−18.47 (3.67)
200	100	−30.08(2.48)	−0.65 (4.58)

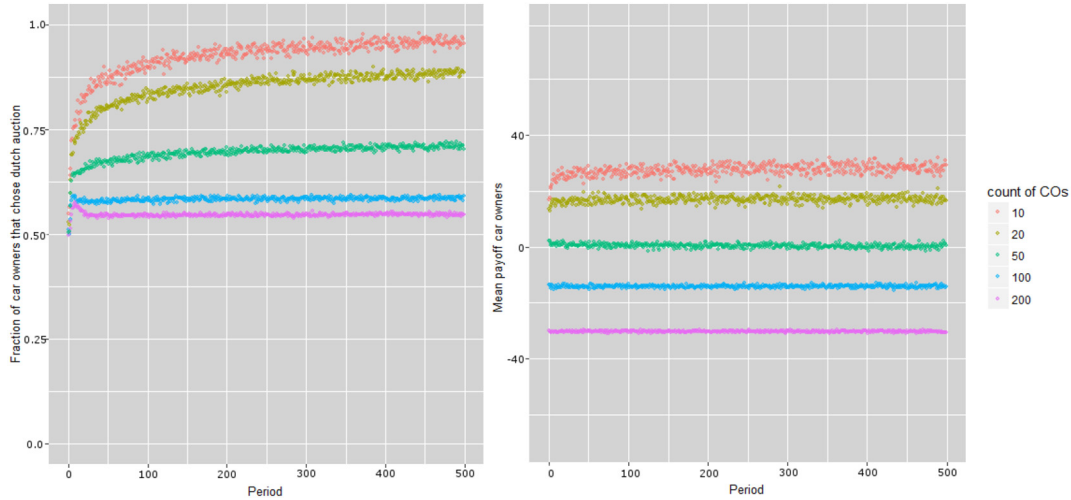

Fig. 6. Fraction of COs who chose Dutch Auction (left) and their mean payoffs (right) with varying CO/NCO ratios.

The mean payoff for NCOs is depicted in Fig. 8 (right). The payoff is negative in almost all cases. But it improves with a higher number of COs, since the possibility to get a seat increases with the number of COs.

3.2.1. Simulation runs with deactivated learning behavior

The two mechanisms Dutch Auction and Fixed Price were investigated separately. In Table 4 the resulting mean payoffs for COs and NCOs (Only Dutch Auction and Only Fixed Price) are shown.

There are higher mean payoffs for COs in the Dutch Auction scenario compared to the Fixed Price scenario. They range from −25.96 to 37.85, whereas in the Fixed Price scenario they range from −33.06 to −0.99. For the NCOs on the other hand there is only a slight difference between the two scenarios, at least for the first three CO/NCOs ratios with CO numbers lower than NCO numbers. Dutch Auctions lead to slightly lower mean payoffs for NCOs: a range of −11.68 to −48.72 opposed to a range of −0.65 to −43.96 in the Fixed Price scenario.

We assume that the better performance of Dutch Auctions for COs and the only slightly lower mean payoffs for NCOs in Dutch Auctions compared to Fixed Price lead to the observed preference for Dutch Auctions by COs in the simulations with active learning behavior.

Also shown in Table 4 are the standard deviations of the mean payoffs. There is a relatively high variation compared to the regular simulations with active learning behavior (Table 3). We assume that

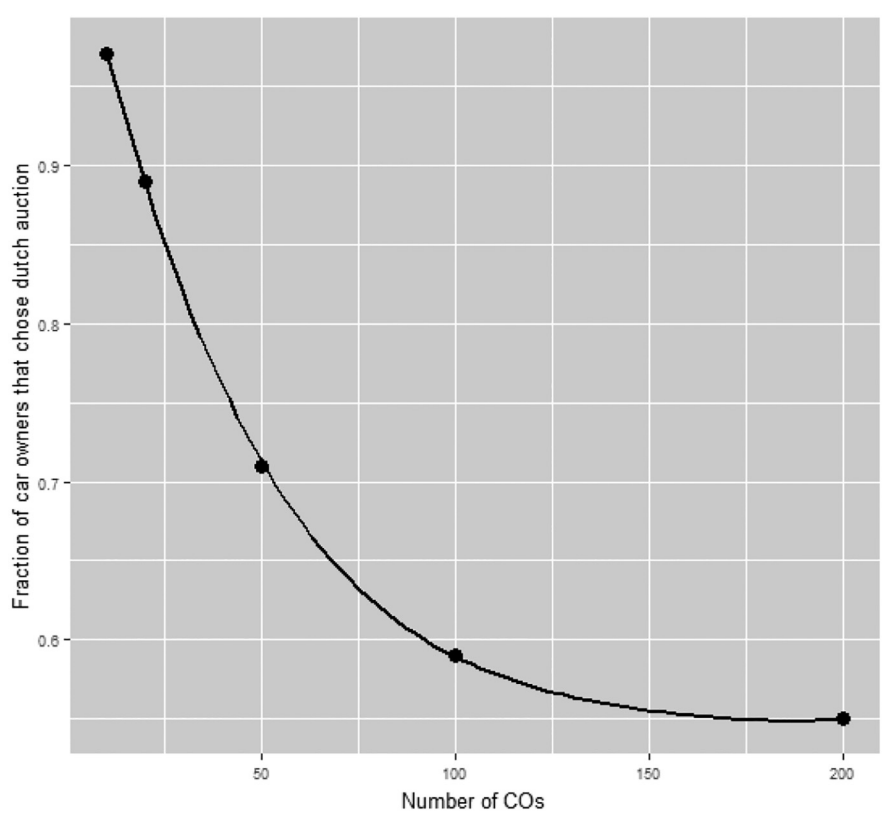

Fig. 7. Mean fraction of COs who chose Dutch Auction with varying CO/NCO ratios.

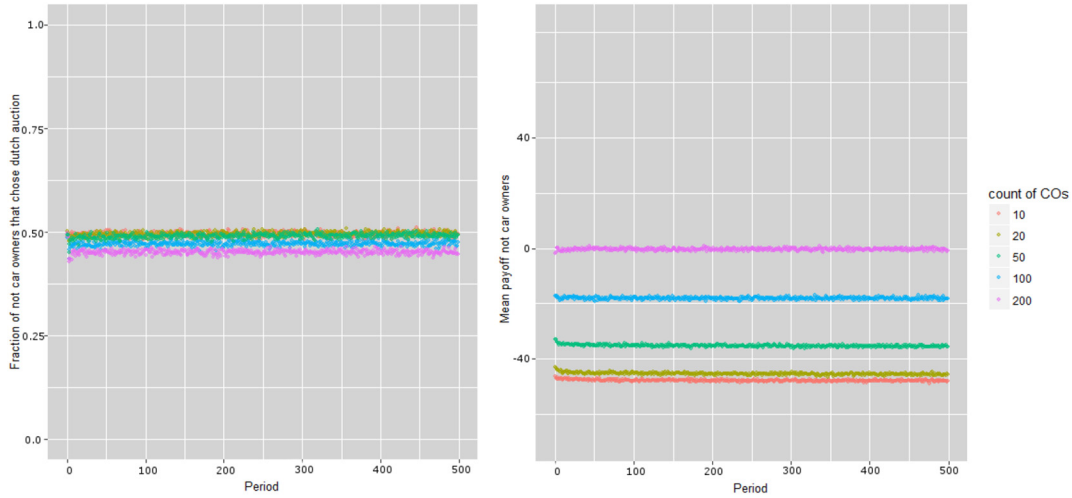

Fig. 8. Fraction of NCOs who chose Dutch Auctions (left) and their mean payoffs (right) with varying CO/NCO ratios.

Table 4
Mean Payoffs for final periods with varying CO counts for runs with deactivated learning behavior ($\overline{\pi}_i$: Mean payoff for COs; $\overline{\pi}_j$: Mean payoff for NCOs; σ: standard deviation.

		Only Dutch Auction		Only Fixed Price	
COs	NCOs	$\overline{\pi}_i(\sigma)$	$\overline{\pi}_j(\sigma)$	$\overline{\pi}_i(\sigma)$	$\overline{\pi}_j(\sigma)$
10	100	37.85 (43.34)	−48.72 (29.88)	−0.99 (9.90)	−43.69 (36.06)
20	100	30.78 (48.72)	−47.26 (31.58)	−1.52 (12.12)	−38.83 (41.29)
50	100	10.95 (56.46)	−41.79 (37.08)	−6.33 (23.53)	−25.33 (48.73)
100	100	−8.03 (57.09)	−29.23 (39.89)	−18.16 (34.22)	−4.66 (46.05)
200	100	−25.96 (51.53)	−11.68 (31.95)	−33.06 (36.33)	−0.65 (27.90)

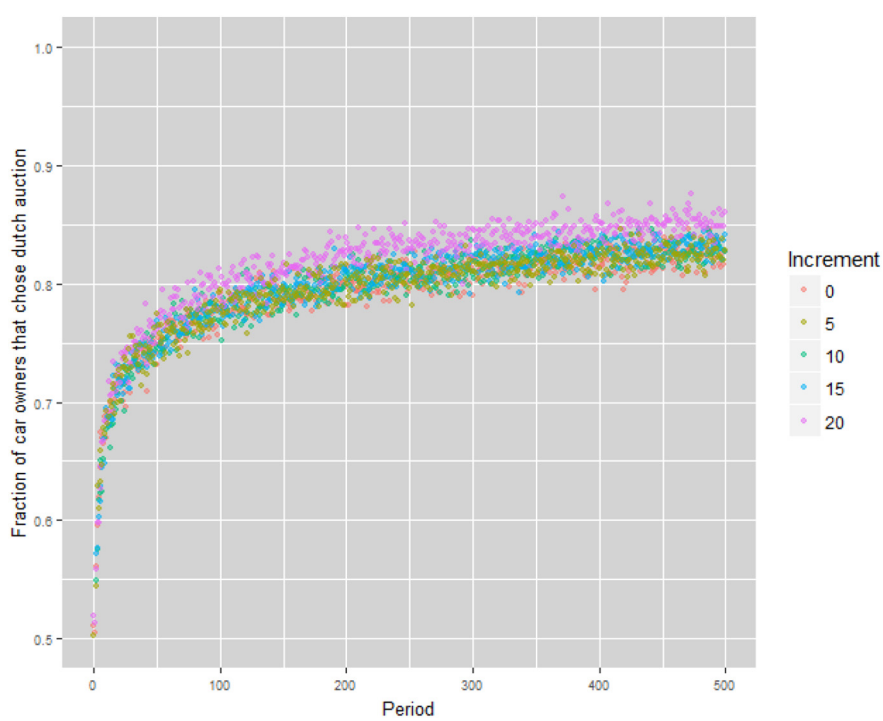

Fig. 9. Fraction of COs, who chose Dutch Auctions in increment scenario.

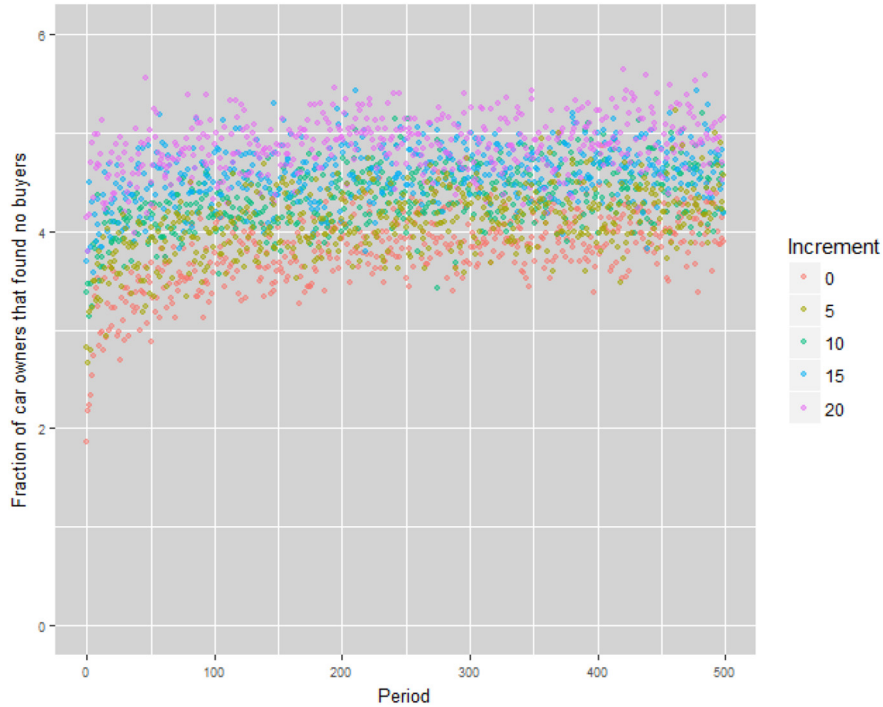

Fig. 10. Empty rides in increment scenario.

this might be caused by the higher number of NCOs available as potential buyers compared to the regular simulation runs, which causes higher variance in the payoffs of completed deals. If we take the 10 to 100 CO/NCO ratio for example we only have about 50 NCOs available in the simulations with learning behavior, whereas in the runs without learning behavior all 100 NCOs are available for the chosen allocation mechanism.

4. A modified fixed price scenario

The design of allocation mechanisms in the present paper - Dutch Auction and Fixed Price - favors Dutch Auctions. Obviously Fixed Price sellers can only achieve selling prices, which are equal to their reservation price, whereas in Dutch Auctions, selling prices are greater than or equal to the reservation prices in the worst case.

Therefore, the scenario has been modified by adding a positive increment to the initial fixed selling prices, which were equal to the reservation prices in the original scenario. This causes strictly positive payoffs for sellers if a trade is successful, which generates higher propensities. The parameters are identical to the ones in the other simulations (see Table 1) and the following increments have been used: 0, 5, 10, 15 and 20.

Nevertheless, the changed scenario doesn't increase the preference for choosing the Fixed Price mechanism. In fact, the opposite is true. With higher increments on the sellers reservation prices the preference for Dutch Auctions increases (see Fig. 9). This is caused by the fact, that with higher selling prices it is less likely to find a matching buyer. This leads to more unsold seats (see Fig. 10), which has a negative effect on the average payoff with a Fixed Price strategy.

5. Conclusion

In the present paper an artificial and simplified scenario is described, which compares two carpooling mechanisms. Under the parameter initializations, which are described in Section 3, where sellers (COs) and buyers (NCOs) had to decide between two mechanisms - Dutch Auction and Fixed Price - the Dutch Auction was clearly preferred by sellers. Buyers slightly preferred the Fixed Price mechanism.

It was shown, that the seller/buyer ratio determine how strong these preferences are. If less sellers were available, then sellers leaned more towards the Dutch Auction. The preference for Fixed Price for buyers was only present if the number of sellers was high, relative to the number of buyers.

Obviously, the presented simulations do not describe a fully realistic scenario of the discussed mechanisms. For example, in further investigations bidding behavior and risk attitudes could be implemented in a more realistic way. Additionally one could use a more dynamic and event based allocation algorithm for matching buyers and sellers. Nevertheless the program is designed to realize these options quite easily.

The Fixed Price mechanism is more common in online platforms. Nevertheless, in this simulated scenarios the Dutch Auction seemed to be more attractive for sellers. Under the given parameter initializations in the presented Reinforcement Algorithm the results show that a Dutch Auction mechanism in the presented carpooling markets increases the number of sellers and therefore the number of offered seats. An increased number of offers then leads to more options for buyers. As a conclusion this might have a growing effect for the carpooling market and its impact as an alternative for transportation. Under these conditions the Dutch Auction performs better than the Fixed Price mechanism. Thus, Dutch Auctions seem to be recommended for use on online platforms.

Author statement

Thomas Pitz: Conceptualization, Project administration, Supervision, Formal analysis, Writing – review & editing, Validation Jörn Sickmann: Writing – review & editing Hasan Alkas: Writing – review & editing Wolf Gardian: Writing – review & editing, Formal analysis, Validation Deniz Kayar: Software, Writing – review & editing, Data curation, Formal analysis, Visualization, Validation.

Appendix A: Pseudocode

Algorithm 2
execAuctions()

```
Algorithm 2: execAuctions()
Data: COs, NCOs, globalTimeStep
Result: Executes auctions recursively that fall into current global time step
begin
    /* Declarations */
    auctionList ← NULL ;                    // List of auctions
    /* Check list sizes */
    if COs.size() == 0 then
        /* All auctions are finished,so all remaining NCOs take the
           train */
        foreach nco ∈ NCOs do
            nco.takesTrain(TRUE);
            NCOs.remove(nco);
        end
    else if NCOs.size() == 0 then
        /* No NCOs left, so finish all auctions */
        foreach co ∈ COs do
            co.auction.finish();
            co.hasFoundBuyer(FALSE);
            COs.remove(co);
        end
    else
        foreach co ∈ COs do
            /* Gather all auctions that fall into current time step */
            if co.auction.stepTime == globalTimeStep then
                auctionList.add(co.auction);
        end
        /* Randomize order of auctions */
        permutate(auctionList);
        /* For each auction search a buyer from the NCOs list */
        searchBuyers(type, auctionList, COs, NCOs);
        /* Increase time step for remaining auctions by their step
           width */
        foreach co ∈ COs do
            co.auction.nextStep();
        end
        /* Recursion */
        execAuctions(COs, NCOs, ++globalTimeStep);
end
```

Algorithm 3
searchBuyers()

```
Algorithm 3: searchBuyers()
Data: auctionList, COs, NCOs
Result: Searches a buyer for each auction in auctionList
begin
    foreach auction ∈ auctionList do
        /* Update current price for current auction */
        currentPrice ← auction.updateCurrentPrice();
        co ← auction.getAuctioneer();
        if currentPrice < co.reservationPrice then
            /* Current price drops below reservation price --> no buyer
               */
            co.setPayoff(0);
            COs.remove(co);
            continue
        /* Randomize order of NCOs */
        permutate(NCOs);
        foreach nco ∈ NCOs do
            if nco.accepts(currentPrice) then
                /* Set payoff and propensities for CO */
                co.setPayoff(currentPrice);
                co.hasFoundBuyer(TRUE);
                co.reinforcementLearning.updatePropensities();
                co.reinforcementLearning.updateProbabilities();
                COs.remove(co)
                /* Set costs and propensities for NCO */
                nco.setCost(currentPrice);
                nco.takesTrain(FALSE);
                nco.reinforcementLearning.updatePropensities();
                co.reinforcementLearning.updateProbabilities();
                NCOs.remove(nco)
        end
    end
end
```

Algorithm 4
agent.reinforcementLearning.updatePropensities()

```
Algorithm 4: agent.reinforcementLearning.updatePropensities()
Data: payoff, strategyChoice
Result: Updated Propensities based on agent payoff and last chosen strategy
begin
    /* Declarations */
    sumsForEachStrategy;
    sumOverAllStrategies;
    /* Initializations */
    sumsForEachStrategy ← Map < strategy, sum >;
    sumOverAllStrategies ← 0;
    /* Update propensities */
    if ultitliy < 0 then
        foreach entry ∈ sumsForEachStrategy do
            if entry.strategy ≠ strategyChoice then
                sumsForEachStrategy(entry.strategy).replace(entry.sum +
                 payoff);
                sumOverAllStrategies ← sumOverAllStrategies + payoff;
        end
    else
        sumsForEachStrategy(strategyChoice).replace(strategyChoice.sum +
         payoff);
        sumOverAllStrategies ← sumOverAllStrategies + payoff;
end
```

Algorithm 5
agent.reinforcementLearning.updateProbabilites()

```
Algorithm 5: agent.reinforcementLearning.updateProbabilites()
Data: sumsForEachStrategy, sumOverAllStrategies, periodIndex,
      countStrategies
Result: Updated cumulative distribution function for each strategy
begin
    /* Cumulative distribution function */
    cdf ← Map < strategy, Pair < lower, upper >>;
    /* Initializations */
    lowerValue ← 0;
    upperValue ← 0;
    currentSum ← 0;
    /* Same probabilities for each strategy in first step */
    if periodIndex = 0 then
        prob ← 1/countStrategies;
        i = 0;
        foreach strategy ∈ sumsForEachStrategy do
            cdf.put(strategy, Pair < i ∗ prob, (i + 1) ∗ prob >);
            i ← i + 1;
        end
    /* Probabilities depending on old payoff sums */
    else
        foreach strategy ∈ sumsForEachStrategy do
            upperValue ← lowerValue + (strategy.sum/sumOverAllStrategies);
            cdf.put(strategy, Pair < lowerValue, upperValue >);
            lowerValue ← upperValue;
        end
    return cdf;
end
```

Algorithm 6
agent.reinforcementLearning.getStrategy()

```
Algorithm 6: agent.reinforcementLearning.getStrategy()
Data: cdf
Result: Get random strategy depending on cumulative distribution
begin
    /* Declarations and Initializations */
    p ← randomDouble();
    strategyChoice ← NULL;
    foreach strategy ∈ cdf do
        /* If random number lies in strategy range choose strategy */
        if strategy.lower <= p & p <= strategy.upper then
            strategyChoice ← strategy;
            return strategyChoice;
    end
end
```

References

Arning, K., Ziefle, M., Mühlhans, H., 2013. Join the ride! User requirements and interface design guidelines for a commuter carpooling platform. In: Marcus, A. (Ed.), Design, User Experience, and Usability. User Experience in Novel Technological Environments https://doi.org/10.1007/978-3-642-39238-2_2.

Arthur, W.B., 1991. Designing economic agents that act like human agents: a behavioral approach to bounded rationality. Am. Econ. Rev. Pap. Proc. 81 (2), 353–359.

Arthur, W.B., 1993. On designing economic agents that behave like human agents. J. Evol. Econ. 3 (1), 1–22. https://doi.org/10.1007/BF01199986.

Barto, A.G., Sutton, R.S., Watkins, C.J.C.H., 1990. Learning and Sequential Decision Making. MIT Press, Cambridge, MA, Learning and Computational Neuroscience.

Bush, R., Mosteller, F., 1955. Stochastic Models for Learning. Wiley, New York (ISBN 1614273197).

Camerer, C., Ho, T., 1998. Experience-weighted attraction learning in coordination games: probability rules, heterogeneity, and time-variation. J. Math. Psychol. 42, 305–326. https://doi.org/10.1006/jmps.1998.1217.

Camerer, C., Ho, T., 1999. Experience-weighted attraction learning in normal form games. Econometrica 67, 827–874. https://doi.org/10.1111/1468-0262.00054.

Chen, S.H., Hsieh, Y.L., 2009. Reinforcement Learning in Auction Experiments. International Association for Research in Economic Psychology (IAREP) and Society for the Advancement of Behavioral Economics (SABE) Joint Conference. Saint Mary's University, Halifax, Canada.

Chmura, T., 2005. Analyse, Modellierung und Simulationen von Routenwahlverhalten. PhD thesis. Fachbereich Physik. Universität Duisburg-Essen, Germany.

Chmura, T., Pitz, T., 2007. An extended reinforcement algorithm for estimation of human behaviour in experimental congestion games. J. Artif. Soc. Soc. Simul. 10 (2). http://jasss.soc.surrey.ac.uk/10/2/1.html.

Chmura, T., Georg, R., Selten, S.J., 2012. Learning in experimental 2 × 2 games. Games Econ. Behav. 76, 44–73. https://doi.org/10.1016/j.geb.2012.06.007.

Cohen, M., Zhang, R., 2017. Competition and coopetition for two-sided platforms. Available at SSRN. https://doi.org/10.2139/ssrn.3028138 https://ssrn.com/abstract=3028138.

Cross, J., 1973. A stochastic learning model of economic behavior. Q. J. Econ. 87 (2), 239–266. https://doi.org/10.2307/1882186.

Einav, L., Farronato, C., Levin, J., Sundaresan, N., 2018. Auctions versus posted prices in online markets. J. Polit. Econ. 126 (1), 178–215.

Erev, I., Roth, A.E., 1998. Predicting how people play games: reinforcement learning in experimental games with unique, mixed strategy equilibria. Am. Econ. Rev. Pap. Proc. 88, 848–881.

Farajallah, M., Hammond, R., Pénard, T., 2016. What Drives Pricing Behavior in Peer-to-Peer Markets? Evidence from the Carsharing Platform Blablacar. Working Paper. .

Galland, S., Knapen, L., Yasar, A., Gaud, N., Janssens, D., Lamotte, O., Koukam, A., Wets, G., 2014. Multi-agent simulation of individual mobility behavior in carpooling. Transp. Res. C 45. https://doi.org/10.1016/j.trc.2013.12.012.

Goldbach, C., Kayar, D., Pitz, T., Sickmann, J., 2019. Transferring decisions to an algorithm: a simple route choice experiment. Transport. Res. F: Traffic Psychol. Behav., 402–417 https://doi.org/10.1016/j.trf.2019.08.011.

Guerci, E., Kirman, A., Moulet, S., 2013. An agent-based model for sequential dutch auctions. Simulation Conference (WSC)., 1707–1718 https://doi.org/10.1109/WSC.2013.6721552.

Guerci, E., Kirman, A., Moulet, S., 2014. Learning to bid in sequential dutch auctions. J. Econ. Dyn. Control. 48. https://doi.org/10.1016/j.jedc.2014.09.029.

Hailu, A., Schilizzi, S., 2004. Are auctions more efficient than fixed price schemes when bidders learn. Aust. J. Manag. 29. https://doi.org/10.1177/031289620402900201 The Australian Graduate School of Management.

Harley, C.B., 1981. Learning the evolutionary stable strategies. J. Theor. Biol. 89, 611–633. https://doi.org/10.1016/0022-5193(81)90032-1.

Kaelbling, L., Littman, M., Moore, A., 1996. Reinforcement learning. A survey. J. Artif. Intell. Res. https://doi.org/10.1613/jair.301.

Kayar, D., "Auctionsimulation" (Version 1.0.0, 2020, August 12). CoMSES Computational Model Library. https://www.comses.net/codebases/a89fd647-1126-4930-a3f6-a5c5145076b3/releases/1.0.0/).

Klemperer, P., 1999. Auction theory: a guide to the literature. J. Econ. Surv. 13, 227–286. https://doi.org/10.1111/1467-6419.00083.

Knapen, L., Yasar, A., Cho, S., Bellemans, T., 2013. Agent-based modeling for carpooling. Data Sci. Simul. Transpo. Res. 259–281.

Kultti, K., 1999. Equivalence of auctions and posted prices. Games Econ. Behav. 27 (1), 106–113. https://doi.org/10.1006/game.1998.0658.

Kultti, K., Virrankoski, J., 2004. Equivalence of auctions and posted prices with many sellers in a location. Finn. Econ. Pap. 17 (2), 79–89.

Moriarty, D.E., Schultz, A.C., 1999. Evolutionary algorithms for reinforcement learning. J. Artif. Intell. Res. 11, 241–276. https://doi.org/10.1613/jair.613.

Pentapalli, M., 2008. A Comparative Study of Roth-Erev and Modfied Roth-Erev Reinforcement Learning Algorithms for Uniform-Price Double Auctions. PhD Thesis. Iowa State University.

R Development Core Team, 2008. R: a language and environment for statistical computing. http://www.R-project.org.

Ricardo, D., 1817. The Principles of Political Economy and Taxation. John Murray, London https://doi.org/10.1017/CBO9781107589421 ISBN 978-0486434612.

Rochet, J., Tirole, J.C., 2003. Platform competition in two-sided markets. J. Eur. Econ. Assoc. 1, 990–1029.

Rochet, J., Tirole, J.C., 2006. Two-sided markets: a progress report. RAND J. Econ. 37, 645–667.

Roth, A.E., Erev, I., 1995. Learning in extensive-form games: experimental data and simple dynamic models in the intermediate term. Games Econ. Behav, Special Issue: Nobel Symposium 8, 164–212. https://doi.org/10.1016/S0899-8256(05)80020-X.

Royal FloraHolland, 2018. Facts and figures. https://www.royalfloraholland.com/en/about-floraholland/who-we-are-what-we-do/facts-and-figures.

Rysman, M., 2009. The economics of two-sided markets. J. Econ. Perspect. 23, 125–143. https://doi.org/10.1257/jep.23.3.125.

Selten, R., Chmura, T., Pitz, T., Kube, S., Schreckenberg, M., 2006. Commuters route choice-behaviour. Games Econ. Behav. 58, 394–406. https://doi.org/10.1016/j.geb.2006.03.012.

Shaheen, S., Stocker, A., Mundler, M., 2017. Online and App-Based Carpooling in France: Analyzing Users and Practices—A Study of Blablacar. UC Berkeley Transportation Sustainability Research Center. .

Teal, R., 1987. Who, how and why, transportation research part a: general. J. Financ. 21, 203–214.

Wang, R., 1993. Auctions versus posted-price selling. Am. Econ. Rev. 83, 838–851.

Wang, R., 1998. Auctions versus posted-price selling. The case of correlated private valuations. Can. J. Econ. 31 (2), 395. https://doi.org/10.2307/136330.

Wickham, H., 2009. ggplot2: Elegant Graphics for Data Analysis.