Internet of Things 13 (2021) 100356

Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.elsevier.com/locate/iot

# Lane and speed allocation mechanism for autonomous vehicle agents on a multi-lane highway

Ellie Lovellette[a,*], Henry Hexmoor[b]

[a] Southern Illinois University Edwardsville, 1 Hairpin Dr, Campus Box 1259, Edwardsville, IL 62025, USA
[b] Southern Illinois University Carbondale, 1230 Lincoln Drive, Engineering A319, Mail Code 4511, Carbondale, IL 62025, USA

ARTICLE INFO

*Article history:*
Received 21 August 2020
Revised 19 November 2020
Accepted 29 December 2020
Available online 3 January 2021

*Keywords:*
Mechanism design
Social utility
Internet of Things
Edge computing
Driverless vehicles
Allocation rule, Vehicle-to-vehicle (V2V) communication

ABSTRACT

Following the arc of automation, it is expected that many driverless vehicles will soon be on the road. The need for autonomous traffic to self-organize necessitates the development of advanced algorithms for unsupervised machine-to-machine negotiation.
Mechanisms rooted in game theory detect and resolve contradictions among agents with differing objectives and enable automated collaboration with a socially beneficial outcome. A case study of cyber-enabled vehicles following private objectives and using private ranked preferences to bid for lane and speed changes on a multi-lane highway illustrates interactions that rely on mechanism design. The presented approach constitutes a strategyproof mechanism and thus induces the truthfullness of bids. A lane position and speed setting allocation rule detects and resolves conflict and provides an option for monetization. A NetLogo simulation implements the methodology described. The model allows for additional resources or resource dimensions. Minor domain specific alterations to the distance operator can translate to many application areas and scenarios providing multiple or multi-dimensional resource allocation for a fee.



## 1. Introduction

The burgeoning Internet of Things (IoT) is a network of interconnected smart devices seeking to automate and improve business processes and various aspects of day-to-day life. As the size and price of electronic components dwindle, manufacturers fortify diverse devices to join the smart, online network. Even though the IoT is currently largely vertical [1] and heterogeneous, the expectation is that in the future smart devices will seamlessly network with each other [2] behind the scenes. The rapid expansion of the IoT and the diversity of its devices lead to issues of interoperability. The necessity for efficient interaction between autonomous devices gives rise to the need for reliable mechanisms for automated machine-to-machine negotiation.

Smart agents in the IoT are cyber-physical systems, integrating computation with physical processes [3]. Each agent is designed to receive sensory data and perform problem solving that produces an output. Cyber-enabled actors, the algorithmically controlled mechanism components of cyber-physical systems [4,5], allow these active machine agents to interact with one another and proactively make decisions and take action. For example, smart vehicles as cyber-physical systems are empowered with agent overlays that provide deliberate decision-making capabilities based on actionable intelligence collected from the multitude of smart sensors the vehicles are equipped with.

* Corresponding author.
*E-mail address:* elovell@siue.edu (E. Lovellette).

https://doi.org/10.1016/j.iot.2020.100356
2542-6605/

A vast multi-agent system like the IoT can greatly profit from localizing operations and bypassing remote control entities in the cloud. By taking advantage of edge computing [6], where data is collected, stored and analyzed at the source, subgroups of agents can form networks that are faster, scalable, and more responsive and can take part in efficient real-time unsupervised negotiation. Exploiting the computational, storage and communication power of modern smart devices while pushing the frontier of computing applications away from a centralized authority and redistributing it to the edge of the network [7] improves scalability, tackles issues of message relay delay, simplifies additional required infrastructure, and provides a local distributed computing environment, which in turn improves real-time performance [8]. Fast and reliable performance is especially important in exceedingly dynamic high-stakes application domains like vehicle traffic.

Smart vehicles (i.e., driverless and cyber-enabled) is an arena for a class of cyber actors that exhibits potential compatibility issues among decisions in cases of e.g., lane selection, platooning, cruising speed control, right of way determination, parking assignment etc. An adequate negotiation mechanism has to identify possible congruence concerns and arrive at a satisfactory resolution of conflicts in favor of a common objective. To assure the safety and predictability of agent behavior, there is a need for the creation of systems of rules and policies that embody desired objectives and govern all actions. It is conceivable that there will be decisions made by machine agents that will be unpopular, at odds with human needs, or rife with ethical issues. For instance, a driverless vehicle must determine the best course of action when confronted with an unavoidable collision [9].

Upcoming changes to the legal framework governing traffic and the rapid advancement of smart vehicle technology will eventually place consumer-owned autonomous traffic agents on the road. To be deployed, such vehicles must overcome certain societal and economic challenges [10] and reach a sufficient level of technological development. Autonomous agents on the road need to communicate and self-organize based on common safety, optimal road usage, individual and shared objectives, availability and utilization of resources, as well as personal and social utility. Regimented versions of this are found in cooperative adaptive cruise control [11] and in platooning techniques [12]. Different algorithms are needed to govern seemingly straight-forward actions in traffic situations like entering and exiting roads, following cars, overtaking, behavior in different types of traffic, deciding right of way, etc.

Lane changing in particular, is an essential but exceedingly challenging task that has been studied from many different angles. A behaviorally sound lane-changing model is not even available for the traditional environment [13]. For connected and autonomous vehicles, examination of the topic is also ongoing. The largest body of research comes from engineering and concentrates on the mechanics of vehicle control during individual lane transitions and on lane change behavior and trajectory planning [14–25–26]. Other popular areas of interest include verifying the safety of maneuvers [27,28–29] and lane change scheduling [30,31–32]. These models are mainly concerned with longitudinal and lateral movement mechanics and the safety of the subject vehicle but tend to ignore the effect of lane-changing on neighboring vehicles and on the flow of traffic. On the algorithmic level, some conceptual collaborative models for lane changes have been proposed. Some concentrate on cooperative sensing - the collaborative sharing of situational awareness to facilitate lane changes [[33,34] or on creating lane assignments based on destination to increase traffic throughput [34,35]. Others study cooperative control - cooperation with immediately neighboring vehicles in the current and target lanes to facilitate a lane change request while avoiding collisions and minimizing the effect on following vehicles [36–39–40]. Lane changing decision modeling for both human drivers and smart vehicles is usually rule-based, utility-based, or game theoretical. Work concerning algorithmic lane changing mechanisms for autonomous and connected vehicles stemming from game theory is still sparse [41]. One framework to model information flow and driver behavior in a connected environment uses different acceleration models to capture the underlying dynamics of car-following behavior and evaluates whether a lane change is beneficial through the acceleration of both the subject vehicle and its surrounding vehicles [42]. Another model uses game theory to model mandatory lane changing decisions [13]. An approach formulating lane changing decision-making as a differential game takes the acceleration of the preceding and following vehicles into consideration to optimize the joint cost and determine a unique and continuous path for the vehicle actuators to track [43]. In a game of incomplete information vehicles use gap selection and interaction through turn signals and lateral moves. Through others' reactions to these queues the model learns in real time to determine optimal timing and acceleration for lane changes [44]. A model based on argumentation augmented with social voting delivers a democratic assignment of requested lane changes on a multi-lane road [45]. Another approach uses a cooperative game of transferable utility in which when finding gaps, vehicles exchange right-of-way for payment, gaining time in the process [46]. A natural extension of that model could be found in mechanism design, where lane changes could be monetized and all players would be winners. This work presents such a model. For individual vehicular agents, the outlined multi-agent game theoretic approach can be used to construct an unsupervised negotiation mechanism to regulate (and monetize) lane and speed changes on a multi-lane highway. The outlined mechanism can be applied to any other field where agents are required to share multiple and/or multi-dimensional resources after a slight modification to the domain-specific operation.

This paper is organized as follows: Section 2 briefly presents the basic game theoretic concepts relevant to the discussion; Section 3 outlines the theoretical foundation for a mechanism for social utility maximizing lane and speed allocation among driverless vehicles on a multi-lane highway and offers an example scenario; Section 4 delineates a test bed environment; Section 5 discusses a prototyped NetLogo simulation of the presented model; Section 6 summarizes the methods presented.

## 2. Game-theoretic mechanism design for smart autonomous agents in the Internet of Things

In a multi-agent system like the Internet of Things, smart devices interact with one another and their decisions affect others. When the moves made by agents working toward an objective are interdependent and pertinent decision attributes include the decisions of others, game theory can be used to formalize the process of reasoning [47]. A game theoretical approach based on mechanism design lends itself well to automated negotiation especially when agents must make decisions on how to share a resource.

### 2.1. Games, strategies, payoffs, and equilibria

In a multi-round game players may maintain private information. In such games of incomplete information, players' payoffs depend on their own type since there is no information about the types of other players. In consultation with its objective, each player agent selects a strategy from an available set. A strategy is a complete algorithmic approach to game play which governs the actions a player takes in every potential situation it can face in the course of the game. The outcome of the game is determined by the strategies played by all participants and payoffs are computed by the game utility function. For the game, or if the game is played in rounds, for each round, the optimal outcome is an equilibrium so that no player would be better off selecting a different strategy. This dominant strategy solution, where each player has no knowledge of and makes no assumptions about the preferences of others but has a unique best strategy independent of the strategies of the rest of the players, may not deliver the optimal payoff for the players but it may lead to an outcome that is desirable. However, games very rarely have a dominant strategy solution and, if it even exists, finding it is a computationally intensive problem. Working backwards, mechanism design, also known as reverse game theory, aims to construct games with dominant strategy solutions.

### 2.2. Mechanism design, strategyproofness, and incentive-compatibility

Mechanism design's goal is to create directives regulating the interaction between agents so that the system in equilibrium represents a desired outcome as defined by the mechanism designer. The laws governing the system produce a collective outcome and are responsible for the system's success or failure [48]. The outcome is often centered around social welfare or profit, or both. This desirable outcome is achieved with the help of a social choice function constructed so that for every combination of individual player types (or whatever input is relevant for the scenario - preferences, judgements, welfare, etc.) there is a Nash equilibrium with the desired outcome. The social choice function itself aggregates the inputs of all players into a single outcome. An incentive-compatible social choice function cannot be strategically manipulated by a player (i.e., no single player can ensure a certain outcome by strategically misrepresenting their type).

The revelation principle of mechanism design [49] posits that any arbitrary mechanism implementing a particular social choice function and its equilibrium outcome (payoffs) can be replicated by an incentive-compatible direct mechanism implementing the same function in which all participants have incentives to reveal information truthfully [[50,51]. In a direct mechanism a player's set of available actions is the set of their possible preferences. A mechanism is characterized as incentive-compatible if every participant achieves the best possible (or at least not worse) outcome just by following their own preferences [52]. The revelation principle is a key piece in finding solutions as it narrows the search field making finding a mechanism easier.

Thus, systems stemming from mechanism design induce truth-telling from players, maintain players' privacy of information, and ensure a desirable social welfare maximizing equilibrium outcome. Today, mechanism design has found many applications in market theory, asset auctioning and allocation, supply chains, taxation, elections, government regulation, politics, logistics, network routing and resource allocation, and transportation, among others. In transportation in particular mechanism design has been applied to congestion theory [53], airport time slot auctions [54], urban road pricing [55], congestion pricing and tradable credit schemes [56,57], tradable permits and ride sharing [58,59–60], commuter assignment [61], ramp control [62,63], traffic flow control [64], parking slot assignment [65], dynamic road pricing [66], dynamic traffic assignment [67], collaborative logistics [68,69], transportation preference elicitation [70], enterprise transportation outsourcing [71]. The next section presents an application of mechanism design to lane changing on a multi-lane road.

## 3. Lane and speed allocation mechanism for autonomous vehicle agents on a multi-lane highway

For the purposes of this work, traffic on a shared road is viewed as a multi-player multi-round game of incomplete information where agents have no private information and no assumptions about others, similar to a Bayesian game [72] without a Common Prior Assumption [73]. Each autonomous vehicle (AV) in the relevant locality is a player in the game and possesses certain private information and preferences related to objectives it maintains. Objectives may vary - e.g., minimizing travel time and/or distance and/or cost, maximizing fuel efficiency, minimizing emissions, lowering lane congestion, etc. An agent has little public information about its neighbors and potentially no information about their private preferences and goals. In order to share the road, a vehicle must select the manner in which it travels. In every round, in concert with its objective, a player makes strategy choices through which it bids for resources, possibly against others. An agent taking up a resource (e.g., a spatial position in a certain lane and driving at a certain speed) means another agent's use of resources may be impeded. Thus, conflict is part of the game and it needs to be resolved in a way that is agreeable or beneficial to players sharing the road. The aim of the game is, for every round, to deliver the highest achievable combined utility for the group of smart vehicles while allowing them to take actions that will satisfy their individual goals. This is achieved through a truth-inducing lane and speed allocation mechanism for AV players on a multi-lane highway where agents bid for lane position and speed setting assignments and receive approval to implement actions that contribute to their individual objectives as well as to a socially beneficial outcome. The model was implemented and verified in NetLogo.

### 3.1. Preliminaries

For a mechanism to be employed, agents have to agree to abide by its rules, accept allocation decisions handed to them, and make payments to cover all applicable cost incurred. In order to present the general case, this work assumes that all AVs on the section of road governed by the mechanism are willing participants and there are no rogue agents on the road acting independently and/or maliciously. Much like modern vehicles equipped with advanced driver safety assist systems, autonomous vehicles will react to sensed data to prevent accidents. So, a rogue agent present in the system undertaking unsanctioned actions should not have an effect large enough to catastrophically undermine the model. Special vehicles (e.g., police, ambulance, fire, etc.) are also excluded, even though the model can be readily extended to accommodate priority agents.

AVs must communicate their situational awareness and agree with everyone else on the sensed data. Since this is information that can be readily extracted from the environment by all participants and outside entities, consensus on the sensed data is implied.

The presented mechanism is a de facto auction, which can be generally implemented in a distributed manner, but in order to maintain privacy of preferences and limit data transfer volume, administration may be left to an appointed manager. A centralized arbiter at the edge of the network can provide computational power, streamlined decision-making, and ensure the reliability of arbitration outcomes. In order to minimize the effect of network topologies with limited connectivity that cannot guarantee the effective propagation of bids to the auctioneer, auctions can be efficiently run within the set of direct neighbors [74,75]. Indeed, as lane changes and speed adjustments in a traffic configuration only affect vehicles in the immediate vicinity, the mechanism manager will evaluate a relatively low number of bids before making a decision, so scalability is provided. Similar to auction algorithms [76], the mechanism can be implemented when in the absence of a dedicated arbiter (e.g., a cell tower base station with coverage over a portion of the road) one agent is assigned or incentivized to accept the additional task of manager. How ever mechanism management is assigned, it is kept local at the edge of the IoT. All agents will register with and abide by the decision of the authority responsible for the section of the road they are currently on.

AVs may maintain an arbitrarily large number of objectives but for simplicity here they only have one main objective which is the basis for decision making. Decisions result in the ranking of desirability of potential actions.

### 3.2. The environment

Consider a multi-lane expressway simulated in the simplest way. Lanes have characteristics pertaining to (i) *maximum speed* and (ii) *minimum speed* with the stipulation that speed limits increase in the higher lanes. Vehicles on the road maintain a governing objective. Relevant characteristics for the agents include their (i) *current speed*, (ii) *preferred speed*, and (iii) a level of *objective emphasis*. The preferred speed and objective emphasis levels determine the target speed and target lane for each player. On the road, each vehicle maintains a required safety distance (buffer zone) from the vehicle directly ahead of it.

### 3.3. The density of agents

The buffer zone requirement puts a natural upper bound on the number of vehicles that can travel in each lane. Depending on lane speed limits, the buffer zone size may fluctuate.

**Lemma 1.** *The maximum number of vehicles n of uniform length c allowed in a single lane segment of length L is* $n = \lfloor \frac{L}{c+d} \rfloor$ *where d is the required safety distance of the lane.*

*Proof.* The space that an agent occupies in the lane is its own length plus the required safety distance. For $n$ agents $nc + nd \le L$ or $n \le \frac{L}{c+d}$. Since $n$ is a positive integer, $n = \lfloor \frac{L}{c+d} \rfloor$. □

**Corollary 1.** *Let a road segment of length L have m lanes with required safety distances* $d_i$ *for each lane,* $i \in (1, \ldots, m)$. *For vehicles of varying length* $c_j$ *with* $j \in (1, \ldots, n)$ *, the maximum number of vehicles allowed on the road segment is* $n = \sum_{i=1}^{m} \lfloor \frac{L-C_i}{d_i} \rfloor$ *where* $C_i$ *is the combined length of allowed vehicles in lane i.*

*Proof.* For a single lane $i$, the combined length of allowed $n_i$ vehicles is $C_i = c_1 + \cdots + c_{n_i}$ . Then $\sum_{j=1}^{n_i} (c_j + d_i) \le L$, or $n_i \le \frac{L-C_i}{d_i}$. Therefore, $n_i = \lfloor \frac{L-C_i}{d_i} \rfloor$. For all lanes combined $n = \sum_{i=1}^{m} n_i = \sum_{i=1}^{m} \lfloor \frac{L-C_i}{d_i} \rfloor$. □

### 3.4. The feasibility of actions

Vehicles sense the environment and make decisions on the feasibility and desirability of potential actions pertaining to speed settings and lane positions. They do this with the help of their governing objective. This objective and the emphasis the vehicle places on it have a direct effect on the way an agent makes decisions and how it perceives the decisions of others. The perception of how much a potential action advances the agent's main goal will result in its quantitative ranking.

Agents in the system make decisions along two dimensions - lane and speed. Thus, consider possible atomic actions for all vehicles representing lane/speed action combinations: (1) *move up and decelerate*, (2) *move up and maintain speed*, (3) *move up and accelerate*, (4) *stay in the current lane and decelerate*, (5) *stay in the current lane and maintain speed*, (6) *stay in the current lane and accelerate*, (7) *move down and decelerate*, (8) *move down and maintain speed*, and (9) *move down and accelerate*. The conditional rule sets pertaining to these actions consider system priority, basic physical limitations, and environmental constraints.

**Algorithm 1:** Feasible Action Derivation

1.1 **if** $LaneUp$ in bounds **and** $LaneUp$ not occupied **then**
1.2 $\quad CanMoveUp =$ true;
1.3 **if** $LaneDown$ in bounds **and** $LaneDown$ not occupied **then**
1.4 $\quad CanMoveDown =$ true;
1.5 $CanMaintainSpeed =$ true;
1.6 **if** $SpaceBehind$ not occupied **or** ($SpaceBehind$ occupied **and** $CurrentSpeed$ of $CarBehind < CurrentSpeed$ of myself) **then**
1.7 $\quad CanDecelerate =$ true;
1.8 **if** $SpaceAhead$ not occupied **or** ($SpaceAhead$ occupied **and** $CurrentSpeed$ of $CarAhead > CurrentSpeed$ of myself) **then**
1.9 $\quad CanAccelerate =$ true;

As seen in Algorithm 1, an agent can potentially move up if the lane above is within the bounds of the environment and the spatial position the agent seeks to occupy in it is not obstructed (lines 1.1–1.2). Not obstructed means the desired space is not currently occupied by another agent and there are no other agents within the buffer zone cone that represents safe space so that potential collisions with decelerating vehicles ahead and accelerating vehicles behind in the target lane are avoided. Similar conditions of bounds and availability apply to the action of moving down (lines 1.3–1.4). Acceleration and deceleration are contingent upon the availability of space ahead or behind coupled with the speed of blocking vehicles. The option to maintain the current speed level is presumed to be always available (line 1.5) but this does not come at the cost of safety. All speed decisions are preempted by a collision avoidance mechanism in which an agent, when distance and speed discrepancy conditions are met, will decelerate or match its speed to the speed of the car directly ahead to prevent a crash.

With the feasibility of separate lane and speed actions ascertained, the AV agent can determine the availability of atomic lane/speed action combinations as defined by the system. Once all feasible atomic actions are identified, they are ranked. The ranking is decided by rule sets for each objective taking into account the target lane and speed, the current speed, the current traffic lane and its speed restrictions, and the physical availability of space as determined by the neighboring agents. All players rank action combinations depending on where their current speed setting and lane position are in relation to their targets.

### 3.5. Agents, preferences, strategies, and outcomes

With the preliminaries established, consider the model's theoretic foundation.

Assume that on a section of road with a designated manager there is a set (including the manager, if applicable) of forward moving self-interested driverless vehicle agents $I$ adhering to safety distance requirements. Also assume that each agent $i \in I$ has a type $\theta_i \in \Theta$ drawn from a set of possible types $\Theta$. A *type* is the private information and preferences of the agent within the context of its dominant objective. The type of an agent is static as long as its objective remains unchanged. Some information in $\theta_i$ is constant - e.g., the make and model and physical dimensions of the vehicle.

In general, and setting aside the possibility to form coalitions, a selfish player makes decisions aiming to maximize its own utility. In every round, a player $i$ uses its objective's rule sets to assign valuations to all currently feasible actions. These valuations over all currently available actions over ranked outcomes represent the highest price the agent is willing to pay to implement the corresponding atomic action. This makes up the main portion of a player's strategy, which is submitted to the mechanism manager in a bid to receive approval to implement the highest valued option next. For example, consider a smart vehicle $i$ whose priority is to minimize its own travel time. The vehicle aims to reach a certain higher target speed, but it currently finds itself in the slowest lane travelling at a speed close to the lane's upper limit. The agent has available outcomes $o_i$ from a set of possible outcomes $\mathcal{O}$. The first and most desirable outcome, $o_1$, would be to move to a lane with a higher speed limit where $i$ can accelerate. Should that action not be available because the mechanism does not grant approval, the vehicle could arrive at outcome $o_2$, in which it accelerates in its current lane. If not allowed, the next best outcome, $o_3$, is to maintains speed and continue travelling in the current lane for the next cycle and then attempt to make changes that lead to a more desirable outcome in the following rounds. Another less beneficial but nevertheless potentially available option is to decelerate in the current lane ($o_4$). There is no lane below, so moving down is not currently possible. Obviously, in accordance with the “get there faster” objective, the vehicle can rank the outcomes' desirability as $o_1 > o_2 > o_3 > o_4$, and the potential utility derived from each outcome follows the same ordering $u_i(o_1, \theta_i) > u_i(o_2, \theta_i) > u_i(o_3, \theta_i) > u_i(o_4, \theta_i)$. This order of utilities for the given outcomes allows the agent to assign a certain valuation $v_i = (v_1, ..., v_4)$ to each of the 4 actions it can take where the valuation of an action is a positive scalar within a mechanism defined range, e.g. [0,1]. Since $o_1$ is the most desirable outcome, the action tied to it, move to a lane up and accelerate, will be valued highest, while decelerating in the current lane will receive the lowest valuation.

**Definition 1.** Let $i \in I$ be an autonomous self-interested agent with type $\theta_i$, where $\theta_i$ contains public and private information about the agent, including its valuation of $n$ potential actions $v_i = (v_1, ..., v_n)$ derived from ranking projected action outcomes within the context of the agent's current governing objective.

Agent $i$'s *strategy profile* $S_i$ is the set of all available strategies $s_i(\theta_i)$.

Agent $i$'s *feasible strategy profile* $FS_i$ is the set of all *feasible strategies* $fs_i(\theta_i)$ as determined by some set of scenario and environment constraints.

Each vehicle $i$ in the designated area of a manager “plays” a feasible strategy $fs_i(\theta_i)$ in the form of a bid at some regular time interval (a cycle). A cycle represents a round of the game made up of a certain number of time slices. In each round, the manager uses part of the time slices to gather the bids, extract the best configuration in its specific area, and signal the approved actions to each player. The remaining time slices are used for implementation of actions, forward progression, adjustment period, and valuation derivation. The strategy profile of the round is used to select an outcome delivering the highest achievable combined utility. The selection is made with the help of a *social choice function*.

**Definition 2.** A *social choice function* $f : FS \to \mathcal{O}$ selects an outcome $o \in \mathcal{O}$ from the set of feasible strategies $fs = (fs_1(\theta_1), ..., fs_I(\theta_I))$ submitted by all $I$ agents.

A social choice function essentially maps each feasible strategy profile to a single outcome. However, note that the social choice function assumes full information about agents and their preferences.

The utility an agent experiences from driving on the mechanism-controlled road is $u_i(o) = v_i(o) - p_i + c_i$ where $c_i > 0$ is some fixed utility attributed to forward progression and $p_i$ is a mechanism specific payment that may have been incurred for implementing the approved action. The approved operation will always be one of the ranked actions submitted by the player.

Conflict could arise when a vehicle's potential lane/speed action would result in a lane position that coincides or interferes with the lane position another vehicle is vying for during the same cycle. Given the position an agent occupies at the beginning of the bidding time slice and the information it submits about intended actions, the projected position at the end of the action implementation time slice can be readily calculated.

Within this mechanism, for each round, every vehicle in the group ranks potential actions along two dimensions - lane and speed. A player is given a choice of $n$ possible (but not always feasible) predetermined actions regarding speed - e.g., decelerate by 5 miles, maintain speed, accelerate by 5 miles, accelerate by 10 miles, etc. Along the lane dimension, there are $m$ theoretically available actions for lane changes - move to a lane down, remain in the same lane, move to a lane up, move up two lanes, etc. The agent determines the valuation of each possible lane/speed atomic action as seen through the prism of its main objective. Therefore, at time slice $t$ the feasible strategy $fs_i^t(\theta_i)$ played by the agent contains information about its current position $pos_i^t$, velocity, physical dimensions, and a lane/speed valuation matrix $v_i^t$ as described in Definition 3.

**Definition 3.** For an autonomous vehicular agent $i$ with mechanism predetermined $m$ possible lane actions and $n$ possible speed actions, the $m \times n$ matrix $v_i^t = \begin{pmatrix} (v_i^t)_{11} & \cdots & (v_i^t)_{1n} \\ \vdots & \ddots & \vdots \\ (v_i^t)_{m1} & \cdots & (v_i^t)_{mn} \end{pmatrix}$ is the *valuation* of all feasible lane/speed action combinations agent $i$ includes in its strategy $fs_i^t(\theta_i)$ for time slice $t$.

Additionally, $\forall (v_i^t)_{xy} \in [0, 1]$, $(v_i^t)_{xy} = 0$ iff either or both lane action $x$ and speed action $y$ are unfeasible, and for all feasible action combinations $xy$ and $wz$ $(v_i^t)_{xy} \neq (v_i^t)_{wz}$, where $x, w \in \{1, ..., m\}$ and $y, z \in \{1, ..., n\}$ and $x \neq w$, $y \neq z$.

In other words, the valuation matrix consists of distinctively valued positive scalars, which signal the current strength of intent a vehicle has to implement the corresponding actions and the maximum price it is willing to pay to do so.

The numeric valuations are assigned according to the current state and objective preferences of the agent. Lane moves are prioritized over speed settings so an agent will always choose to make a change in the direction of the target lane over staying in the current lane over moving in the opposite direction. The speed preference is set according to its feasibility given the potential lane action, lane speed constraints, and target speed setting. Thus, feasible actions that bring the vehicle closer to its target lane or keep it in the target lane receive higher valuations. Multiple atomic actions that result in the same lane position are uniquely ranked according to their potential to reach or keep the target speed. Ranking algorithm details are omitted in the interest of brevity.

Note that even though Definition 3 contains language pertaining to the driverless vehicle application domain, the valuation in its present form can be used for bids on any two-dimensional divisible resource. As a matter of fact, more dimensions can be added by adding matrices, or the valuation can be scaled down to a vector.

### 3.6. Conflict and conflict-free action allocation

Using the timestamped strategies submitted by all agents in its area, the manager extracts the relevant information and calculates a position matrix $pos_i^{t'} = \begin{pmatrix} (pos_i^{t'})_{11} & \cdots & (pos_i^{t'})_{1n} \\ \vdots & \ddots & \vdots \\ (pos_i^{t'})_{m1} & \cdots & (pos_i^{t'})_{mn} \end{pmatrix}$. Each element of $pos_i^{t'}$ represents the potential physical position (relative or absolute depending on the type of positioning employed) a vehicle would occupy at the end of the implementation time slice $t'$ should the corresponding plausible lane/speed action combination from the valuation matrix be granted. Thus, for each agent $i$ there is a one-to-one correspondence between elements in $v_i^t$ and $pos_i^{t'}$. Relevant to the application domain, every element of $pos_i^{t'}$ is in the form $(Lx, f, r)$, where $x$ is the lane number, and $f$ and $r$ are the projected spatial positions of the front and rear bumpers of the vehicle respectively.

The mechanism determines assignment allocation for each vehicle along both the lane and speed dimensions. It calculates the set $L$ of all possible assignments of lane actions for all vehicles. Since every vehicle can be assigned one of $m$ lane actions, then $L = \{l_{11}, ..., l_{1m}; ...; l_{I1}, ..., l_{Im}\}$ where $l_{ij}$ is an $m$-dimensional vector and $i \in I$. Also, $\forall k \in \{1, ..., m\}$ $(l_{ij})_k = 1$ iff $k = j$, $(l_{ij})_k = 0$ if $k \neq j$ and $\sum_{k=1}^{m} (l_{ij})_k = 1$. This means that each vector $l_{ij}$ has a single non-zero element and that element is equal to 1 (or can be replaced with a multiplier the mechanism designer puts into place to scale payment as appropriate). Similarly, the set of all possible assignments of speed actions is $S = \{s_{11}, ..., s_{1n}; ...; s_{I1}, ..., s_{In}\}$ with the same characteristics – $s_{ij}$ is an $n$-dimensional vector, and for $\forall k \in \{1, ..., n\}$ $(s_{ij})_k = 1$ iff $k = j$, $(s_{ij})_k = 0$ if $k \neq j$ and $\sum_{k=1}^{n} (s_{ij})_k = 1$. In other words, each player is assigned a single lane and a single speed action.

Next, while calculating the maximum social welfare of the group, the manager detects conflict for lane positions. Conflict occurs when the result of potential actions, when undertaken by a pair of vehicles would result in lane positions that are incompatible, i.e. they cannot occur at the same time because they would either occupy overlapping space or violate mechanism spacing constraints.

**Definition 4.** An outcome $(l, s) = (l_1, s_1; \ldots; l_I, s_I) \in \mathcal{O}$ with $l_i \in L$ and $s_i \in S$ is a *lane and speed allocation* where $\forall i, j \in I$, $i \neq j$ and agent $i$ is not in conflict with agent $j$, i.e.$|pos_i^{t'} s_i l_i \ominus pos_j^{t'} s_j l_j| \geq d$ where $\ominus$ is an operation calculating the distance between two positions in the same lane and $d$ is a mechanism designated required safety distance between vehicles in the same lane.

For a given lane and speed allocation $(l, s)$, $pos_i^{t'}(l, s) - pos_j^{t'}(l, s) = pos_i^{t'} s_i l_i \ominus pos_j^{t'} s_j l_j = (Lx, \ f_i, r_i) \ominus (Ly, \ f_j, r_j) =$
$$\begin{cases} \infty & \text{if } x \neq y \\ f_j - r_i & \text{if } x = y, \ f_j \geq f_i \\ f_i - r_j & \text{if } x = y, \ f_j < f_i \end{cases}$$
Therefore, if two or more vehicles are vying for coinciding or intersecting spatial positions on the road, the manager must calculate the maximum attainable utility, appoint the winner of the debatable position and assign the losing agent(s) conflict-free actions of lesser valuation. The resulting lane and speed allocation contains one vector from $L$ and one vector from $S$ for each agent.

Note that the operation $\ominus$ is specific to the application domain. It has to be redefined as appropriate to use in other fields.

The optimal allocation outcome $(l, s) \in (L, S) \in \mathcal{O}$ extracted by the social choice function where all agents submit their true valuations is obtained by solving $(l, s) \ = argmax_{l \in L, s \in S} \sum_{i=1}^{I} v_i^t(l, s) \ = argmax_{l \in L, s \in S} \sum_{i=1}^{I} v_i^t s_i l_i$ where $\forall pos_i^{t'}(l, s)$ is conflict free within the allocation.

The maximum exists, because solving for *argmax* delivers a non-empty set since the status quo always exists.

**Observation 1.** Since $(l, s)$ is the optimal truth telling conflict-free allocation, $\nexists(l', s')$ that improves social welfare, i.e. $\sum_{i=1}^{I} v_i^t(l, s) \geq \sum_{i=1}^{I} v_i^t(l', s')$.

### 3.7. Prices and payouts

For each agent the optimal obtainable allocation result $v_i^t(l, s) = v_i^t s_i l_i$ is the maximum valuation for a combination of lane/speed actions at bidding time slice $t$, the implementation of which results in a projected position that is either not disputed or is won by the agent. The utility the agent receives is $u_i^t(l, s) = v_i^t(l, s) - p_i^t + c_i$ where $c_i$ is constant and $p_i^t$ is the potential cost incurred by agent $i$ for implementing the mechanism approved action at the end of the implementation time slice. $p_i^t$ is determined by the difference in social welfare when the agent is not present in the group and when it is.

**Definition 5.** The *mechanism specific price* $p_i^t = \sum_{j=1, j\neq i}^{I} v_j^t(l, s)_{-i} - \sum_{j=1, j\neq i}^{I} v_j^t(l, s) \geq 0$, where $(l, s))_{-i}$ is the optimal conflict-free allocation for the group without taking agent $i$'s presence into consideration and $(l, s)$ is the optimal conflict-free allocation for the whole group (but for the term the position assigned to agent $i$ is de facto unavailable to all other agents).

In a matter of speaking, the price an agent pays is the cost of the agent inflicting itself on the remaining group, the "damage" it imposes on its society. In mechanism design, this is known as the Clarke pivot rule [77]. The price is always non-negative because an agent's presence and reported preferences may disrupt the configuration of the remaining agents or may be completely independent of it. In other words, if a player changes the combined utility of the remaining group with its presence, it pays a price.

**Definition 6.** The *utility* of an agent for a time slice $t$ for and allocation $(l, s)$ is $u_i^t(l, s) = v_i^t(l, s) - p_i^t + c_i = v_i^t(l, s) - (\sum_{j=1, j\neq i}^{I} v_j^t(l, s)_{-i} - \sum_{j=1, j\neq i}^{I} v_j^t(l, s)) + c_i = \sum_{i=1}^{I} v_i^t(l, s) - \sum_{j=1, j\neq i}^{I} v_j^t(l, s)_{-i} + c_i$

### 3.8. Lying as a strategy

Autonomous vehicle agents were assumed to be rational and selfish, so theoretically there would be nothing stopping them from trying to gain an edge by falsifying the time slice bid they submit to the manager. Misrepresenting physical information is futile as it can be readily sensed and penalized but what about lying about valuation? Can an agent purposefully misreporting valuations fraudulently win the lane/speed action, diminish its incurred mechanism-specific price, or drive up its own utility?

**Proposition 1.** No agent can be better off by lying.

*Proof.* Should an agent $i$ lie about its true valuation $v_i^t$ in order to try to gain an edge and instead submit a valuation $(v_i^t)'$ while all other agents submit their true preferences, an alternative allocation $(l', s')$ will emerge.

The new utility of agent $i$ thus becomes $(u_i^t)'(l', s') = \sum_{i=1}^{I} v_i^t(l', s') - \sum_{j=1, j\neq i}^{I} v_j^t(l, s)_{-i} + c_i$. However, it was already shown in Observation 1 that the optimal allocation where all agents divulge their true valuations maximizes social welfare. Thus, since $\sum_{i=1}^{I} v_i^t(l, s) \geq \sum_{i=1}^{I} v_i^t(l', s')$ and the second and third utility terms are common and constant for each agent $i$, it follows that no agent can be better off by lying about its true preferences. □

**Observation 2.** Truth telling in this case presents a dominant strategy since it maximizes the utility of an agent regardless of what strategies the other agents play. This is beneficial, given that agents have no knowledge of the types and preferences of others. Since all agents are best served by telling the truth, the game has a dominant-strategy equilibrium, which is also a Nash equilibrium.

### 3.9. The mechanism

A social choice function maps the true preference of a group of agents to an outcome and depends on agents telling the truth. The social choice function (previously presented in Definition 2) $f : FS \rightarrow \mathcal{O}$ selects an outcome $(l, s)$ given full information about agents' preferences $fs = (fs_1(\theta_1), ..., fs_I(\theta_I))$ where $fs \in FS = FS_1 \times \cdots \times FS_I$.

A mechanism utilizes an outcome function to produce a social utility maximizing allocation assignment that depends on the preferences of the agents. The preference information however, is private and a self-interested agent may misreport it in an effort to achieve a better outcome for itself. Thus, since truthfulness is not guaranteed, a mechanism is constructed to map the *reported* type of an agent to a desired outcome. In particular, a mechanism that returns the same outcome that would have been reached had the agents been telling the truth about their preferences is said to implement the social choice function.

**Definition 7.** A mechanism $\mathcal{M}(S, g)$ *implements the social choice function* $f$ if there is an equilibrium strategy profile $s^*$, s.t. $g(s^*) = f(fs)$, $\forall fs \in FS$.

In other words, each agent is afforded a set of strategies $S_i$ from which it selects one. The strategy may reflect the truth, but it also may not. The strategy set for all agents $s = (s_1(\theta_1), ..., s_I(\theta_I)) \in S = S_1 \times \cdots \times S_I$ is used by the mechanism outcome function $g : S \rightarrow \mathcal{O}$ to produce a result. If no agent has an incentive to change its chosen strategy, the strategy set is an equilibrium $s^* = (s_1^*(\theta_1), \ldots, s_I^*(\theta_I))$. If the outcome of that equilibrium produced by the outcome function is the same as the outcome produced by the social choice function, the mechanism $\mathcal{M}(S, g)$ implements the social choice function.

**Definition 8.** A mechanism $\mathcal{M}(S, g)$ is a *direct revelation mechanism* if $S_i = FS_i$ for $\forall i$ and $g(fs) = f(fs)$ for $\forall fs \in FS$.

Definition 8 posits that a direct revelation mechanism does not put a restriction on the strategy space of the players and achieves the same outcome regardless of the truthfulness of played strategies. Recall that the Revelation Principle of mechanism design postulates that if the full information social choice function can be implemented by a mechanism, the mechanism is direct and every agent has an incentive to reveal its true preferences [50,51]. In that case, the social choice function is said to be *incentive-compatible*. If a mechanism is characterized as incentive-compatible, every participant achieves the best possible outcome just by following its own true preferences.

**Definition 9.** A direct revelation mechanism $\mathcal{M}(S, g)$ implements an *incentive-compatible* social choice function if it has a dominant strategy equilibrium $s^* = (s_1^*(\theta_1), \ldots, s_I^*(\theta_I))$ s.t. $s_i^*(\theta_i) = fs_i(\theta_i)$ for $\forall i \in I$.

**Definition 10.** A direct revelation mechanism $\mathcal{M}(S, g)$ where the truth is a weakly dominant strategy for each player regardless of what the remaining players do implements a *strategy-proof* social choice function.

**Proposition 2.** Mechanism $\mathcal{M}(FS, g)$ where $g : FS \to (L, S)$ for $fs = (fs_1(\theta_1), \ldots, fs_I(\theta_I)) \in FS$ is incentive-compatible, strategy-proof and induces truth telling of agents.

*Proof.* Mechanism $\mathcal{M}(FS, g)$ does not restrict the strategy space of agents and delivers the same outcome as the social choice function so it is a direct revelation mechanism.

It was already shown in Proposition 1 that no agent can be better off by lying about its lane/speed plausible action valuation, so the mechanism induces truth-telling of agents.

Per Observation 1 truth telling was shown to be a weakly dominant strategy for each player regardless of what others do. The game thus has a dominant strategy equilibrium coinciding with the true valuations of all agents. Therefore $\mathcal{M}(FS, g)$ implements an incentive-compatible strategy-proof social choice function. □

The presented mechanism $\mathcal{M}(FS, g)$ is a variation of a Vickrey-Clarke-Groves mechanism [77,78,79]. It implements an efficient outcome in dominant strategies. The dominant strategy of each player is the truth and it is independent of the choice of strategy of the remaining players. The Clarke pivot rule [77] ensures that each player is charged its externality, namely the difference in social welfare when the player is absent and when the player is present. Since all valuations are non-negative, the players always get a net positive utility and the mechanism always charges a non-negative price. This makes the mechanism a win-win game - players get to undertake desirable actions and gain more than they pay and the mechanism receives a net positive payment while providing a service.

### *3.10. Example scenario*

Consider the following simplified real world scenario as shown in Fig. 1 where three vehicles travel in immediate vicinity of one another on a three lane highway under the authority of the same mechanism manager. Vehicle1 drives in the fastest lane at the speed limit for the lane, vehicle 2 travels in the middle lane at its target speed, and vehicle 3 is in the slowest lane at a speed below its target. Each vehicle is 15 feet long. For safety reasons, the mechanism enforces a minimum distance of two car lengths (30 feet) between vehicles in the same lane. Additionally, the current governing objective of $v_1$ and $v_3$ is minimizing travel time while $v_2$'s main objective is travelling in a minimally congested lane.

Let the mechanism afford all agents three potential actions for lane changes - move to a lane up, stay in the current lane, move to a lane down; and three potential actions for speed changes - decelerate by 5 miles, maintain speed, accelerate by 5 miles. For simplicity, for each cycle (round of the game) the mechanism accepts bids and evaluates them and the vehicles implement the granted actions in a single time slice. The time slice duration for this example is set at 5 seconds and the time required for data transmission is disregarded. In this scenario, the action valuation matrices the three vehicles formulate in accordance with their main objectives can look as follows (lane changes are recorded in the rows and speed changes in the columns):

$$v_1^t = \begin{pmatrix} 0 & 0 & 0 \\ 0.7 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \; v_2^t = \begin{pmatrix} 0.2 & 0.5 & 0.4 \\ 0.7 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \text{ and } v_3^t = \begin{pmatrix} 0 & 0.8 & 1 \\ 0.1 & 0.4 & 0.5 \\ 0 & 0 & 0 \end{pmatrix}.$$

Since vehicle 1 is travelling in the fastest lane, all options to move to a faster lane (row 1 in the $v_1^t$ valuation matrix) are excluded by assigning them a valuation of 0. The exclusion can be made by the manager or the vehicle. The two best available choices for this vehicle according to its governing objective are to either maintain speed, represented by element $(v_1^t)_{22}$, or decelerate by 5 miles in the same lane, $(v_1^t)_{21}$. The vehicle is already travelling at the speed limit for the lane so the option to accelerate, $(v_1^t)_{23}$, is unavailable. Since the lane down has a maximum speed limit much lower than 5 miles from the agent's current speed, the move to a that lane is currently unavailable until the vehicle decelerates sufficiently. Thus all options in row 3 of the valuation matrix of vehicle 1 are turned off by marking them with 0 s. Vehicle 2's preferred course of action, $(v_2^t)_{22}$, is to remain in its current lane and maintain velocity, since that is its preferred speed. Decelerating in the same lane is the second choice because the new velocity is still close to the preferred speed and the lane is not as congested as the lanes on either side. Moving to the lane up is a less desirable option and delivers lower valuations as shown in the first row of the $v_2^t$ valuation matrix. Moving to the lane down is currently unavailable to vehicle 2, since its speed is too high. Several 5 mph decelerations have to take place before the alternative becomes acces-

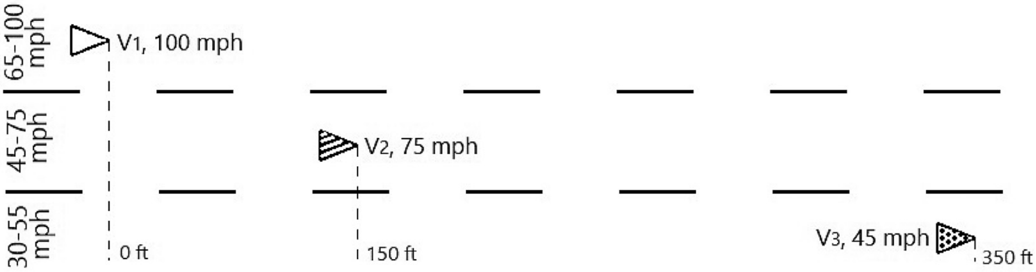


**Fig. 1.** Mechanism Design Among Driverless Vehicles on a Highway Example.

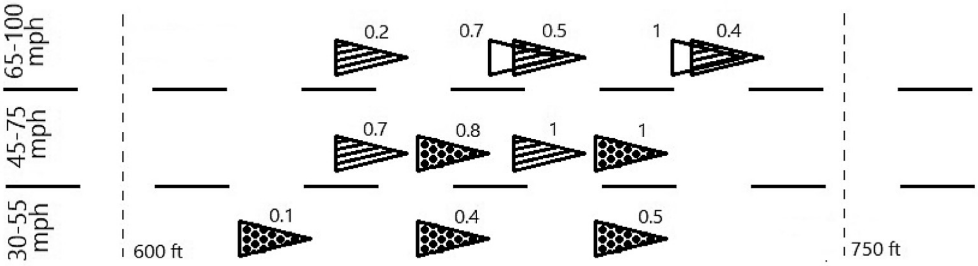


**Fig. 2.** Mechanism Design Among Driverless Vehicles on a Highway Example – Potential Positions at the End of the Implementation Time Slice.

sible. Similarly, vehicle 3′s top choice is to move up a lane and increase speed, $(v_3^t)_{13}$. Lesser available options are ranked accordingly.

Without restriction of generality, assume that the manager is a cell tower with coverage over the road. The manager receives the timestamped bids, unpacks them and calculates the potential position matrices $pos_i^{t'}$ for every feasible action (with valuation greater than 0) of every agent. Distance calculations can be made with the help of positional road markings, GPS, road sensors, etc. The potential agent locations allow the manager to determine positional conflict between vehicles should a lane/speed action combination be allowed to be implemented. For simplicity, the minor distance loss in forward advancement incurred by an agent changing lanes is disregarded. For the example, distance calculation results are rounded up to the nearest whole number. The front of the last vehicle in the group (vehicle 1) can be considered to be relative position 0 as illustrated in Fig. 1. The top valuation of the agent is to remain in its current lane and maintain its speed. At its present velocity of 100 mph $v_1$ would travel 733 feet within the 5 seconds of the time slice. As the length of the vehicle is 15 feet, $v_1$ would occupy the space between 733 and 718 feet ahead of its current location in lane 3, should it be allowed to implement its top choice actions. Similarly, all other non-zero valuations will receive a relative potential position. Thus the resulting positional matrices are:

$$pos_1^{t'} = \begin{pmatrix} -1 & -1 & -1 \\ L3,\ 697, 682 & L3, 733, 718 & -1 \\ -1 & -1 & -1 \end{pmatrix} \text{ corresponding to } v_1^t = \begin{pmatrix} 0 & 0 & 0 \\ 0.7 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

$$pos_2^{t'} = \begin{pmatrix} L3, 663, 648 & L3, 700, 685 & L3, 737, 722 \\ L2, 663, 648 & L2, 700, 685 & -1 \\ -1 & -1 & -1 \end{pmatrix} \text{ to } v_2^t = \begin{pmatrix} 0.2 & 0.5 & 0.4 \\ 0.7 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \text{ and}$$

$$pos_2^{t'} = \begin{pmatrix} -1 & L2, 680, 665 & L2, 717, 702 \\ L1, 643, 628 & L1, 680, 665 & L1, 717, 702 \\ -1 & -1 & -1 \end{pmatrix} \text{ to } v_3^t = \begin{pmatrix} 0 & 0.8 & 1 \\ 0.1 & 0.4 & 0.5 \\ 0 & 0 & 0 \end{pmatrix}.$$

Potential spatial positions with their valuations are shown in Fig. 2.

The mechanism calculates the maximum attainable social welfare when all conflict is resolved. The highest possible collective utility of 3 is unachievable since the distance in top valued potential positions for $v_2$ and $v_3$ would violate the built-in mechanism safety distance of 30 feet. The most social welfare that can be extracted from this configuration is 2.7. Conflict is resolved and as a result, $v_1$ and $v_3$ are allowed to implement their top choice of actions and $v_2$ receives approval for its second choice. The optimal allocation becomes

$$(l, s) = \left( \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}; \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}; \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right).$$

Indeed, this is the best allocation. Of the 50 unique allocation candidates, there are 19 viable conflict-free configurations. Aside from the winning allocation, the remaining 18 result in combined group utility ranging between 1.3 and 2.5.

Next, given the optimal allocation, the mechanism calculates if it's owed any payment for the round.

Price is determined according to the formula presented in Definition 5. Vehicle 1 pays nothing since in this cycle it does not inflict itself on anyone - the best possible utility of the group for an alternative action allocation without it $(s, l)_{-1}$ is 1.7 and the utility of the allocation determined by the mechanism with the utility of vehicle 1 taken out is also 1.7. Similarly, $p_2^t = 2 - 2 = 0$. However, vehicle 3 moving up a lane and essentially forcing vehicle 2 to decelerate causes some externality on the system. The price vehicle 3 pays is

$$p_3^t = \sum_{j=1, j\neq 3}^{3} v_j^t(l,s)_{-3} - \sum_{j=1, j\neq 3}^{3} v_j^t(l,s) =$$

$$= \begin{pmatrix} 0 & 0 & 0 \\ 0.7 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0.2 & 0.5 & 0.4 \\ 0.7 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

$$- \left[ \begin{pmatrix} 0 & 0 & 0 \\ 0.7 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0.2 & 0.5 & 0.4 \\ 0.7 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \right] =$$

$$= 2 - 1.7 = 0.3$$

The manager relays the approved actions to all agents and collects non-zero payment amounts. Actions are implemented and after a short adjustment period using up the remaining time slices, the process restarts for the next cycle.

For simplicity, the constant utility term $c_i$ is disregarded. Therefore, reporting their true valuations resulted in individual utilities for the 3 players of $u_1^t = 1 - 0 = 1$, $u_2^t = 0.7 - 0 = 0.7$, and $u_3^t = 1 - 0.3 = 0.7$. Now consider an attempt by one of the agents to improve its utility by misreporting preference values. Vehicle 2′s governing objective is travelling in less congested lanes, so maybe lying about its valuation for decreasing speed in the same lane will be one way to avoid the necessity to make room for vehicles changing into its lane directly ahead of it. Reasoning this way a rational agent could hide its true valuation for the action, 0.7, and report it as, for instance, 0.1. All else remaining equal, the maximum attainable social welfare of the system drops to 2.5 in which both vehicles 1 and 2 maintain their speed and lane, and vehicle 3 is forced to implement its third choice, $((v_3^t)_{23} = 0.5)$, namely to continue travelling in its current lane but accelerate by 5 miles until the next time slice. Vehicle 3 incurs no cost for this action because the allocation for the remaining two vehicles does not change when agent 3 is not present in the system. Vehicle 1 travels conflict-free during the time slice, so its cost is also zero. However, vehicle 2 pays a price for misrepresenting its valuation and its lack of flexibility. The cost incurred by it is $p_2^t = 2 - 1.5 = 0.5$ and its utility drops to 0.5 (the true valuation of 1 less the price to take the approved actions), so, as shown above, lying not only does not improve outcomes, it may leave a vehicle worse off.

An attempt to achieve certain desired actions by lying about the availability of alternatives, e.g. agent 2 claiming that decelerating in its lane is unfeasible by assigning it valuation 0, can be easily detected and assigned a steep monetary penalty as the manager has situational awareness and knows what actions are available to all of its players.

Finally, a player might consider lying to try to decrease the cost it may incur for the right to implement certain actions. In the example, consider agent 3 misrepresenting its valuation for move up/accelerate as 0.9 instead of 1. The decrease is small enough to still convey the strong desirability of the actions but could potentially decrease the price. However, the cost incurred by a player corresponds to the externality it causes on the system and is calculated disregarding the agent's own report of the importance of actions. Under these conditions, the maximum achievable social welfare is 2.6 (1 from $v_1$, 0.7 from $v_2$, and 0.9 from $v_3$). Both agents 1 and 2 have zero payments, but agent 3 pays 0.3 as before. Undervaluing actions to decrease payment is thus futile and agents are again best served by telling the truth. This is not surprising, since, as shown in Proposition 2, the mechanism is strategy-proof and incentivizes truth telling.

The illustrative example presented here was verified by the NetLogo simulation (see Section 5) as shown in Fig. 3.

Note that in the presented example scenario even though social utility is maximized, $v_3$ will accelerate and pull in in front of $v_2$ which will decelerate per mechanism approved action. The actions are allowed by the mechanism because the constraint specifying the minimum safety distance between agents on the road is satisfied. However, this will still leave a 20 mph difference between the agents' velocities and vehicle 2 will ultimately be forced to decelerate more during the adjustment period between rounds to avoid a collision. A potential refinement of the mechanism will not only detect an agent inflicting an externality on the system but also the degree of its severity. The mechanism will either charge more for inconveniencing other players to a larger extent or adjust buffer zone requirements based on speeds.

## 4. Empirical model

A test bed was created to implement and verify the mechanism.

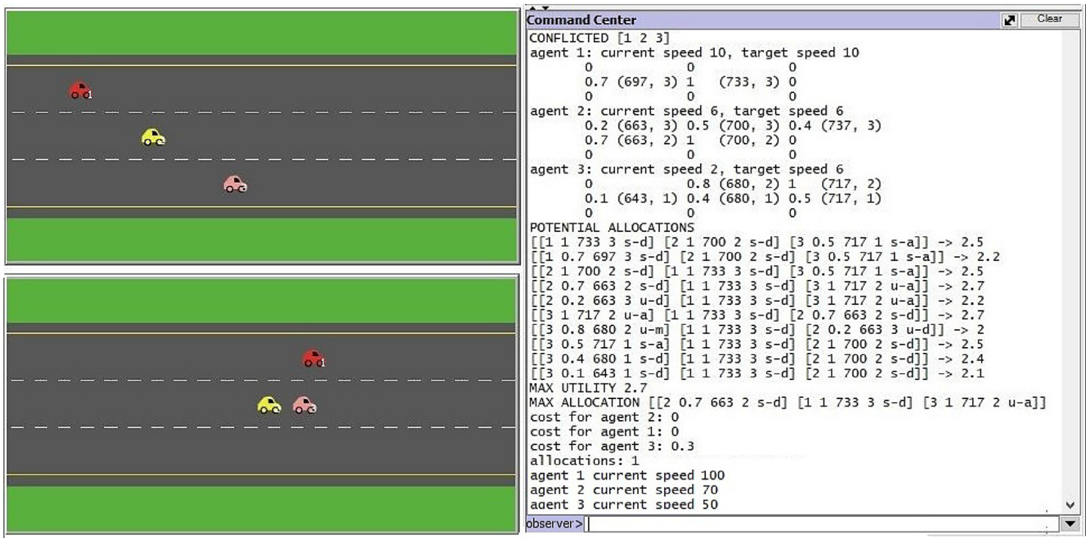


**Fig. 3.** Mechanism Design Among Driverless Vehicles on a Highway Example – NetLogo Simulation Results.

**Table 1**
Lane Characteristics of a Simulated Three-Lane Highway.

| | Emission Levels (1–10) | Maximum Speed (1–10) | Minimum Speed (1–10) |
|---|---|---|---|
| Lane 3 | 10 | 10 | 7 |
| Lane 2 | 7 | 6 | 4 |
| Lane 1 | 4 | 3 | 1 |

### 4.1. Environment and agent characteristics

As outlined before, the simulated expressway has lanes with minimum and maximum speed characteristics. The test bed adds another attribute, emission level. All three attributes were assigned simplified valid ranges with values between 1 and 10. For an example three-lane highway, as shown in Table 1, the top lane, Lane 3, is characterized by the maximum overall speed but also the maximum emission level. At the bottom, Lane 1 is the slowest but most emission-friendly lane.

The vehicle agents' current and preferred speed attributes are valued between 1 and 10, with 10 being the highest. The objective emphasis attribute takes values between 1 and the number of lanes being simulated, with the higher values commensurate with the importance the vehicle assigns to its governing objective. In the test bed the objective emphasis and preferred speed attributes are randomly generated within the feasible ranges.

An assumption is made regarding prevailing laws governing the travel of vehicles from one lane to another. Without restriction of generality, it is assumed that once a vehicle starts to travel to a different lane the movement will be largely perpendicular to the vehicle's current position. Trajectory generation and tracking are not a subject of this algorithmic model.

### 4.2. Agent objectives, targets and atomic action derivation

This case study defines three potential objectives: agents may choose to prioritize minimizing personal travel time (TT), reducing lane congestion level (CL), or reducing global emissions (EL). By maintaining an objective, vehicles on the road are working against some or all of the remaining objectives. For example, a vehicle focused on minimizing its travel time will likely select to drive in a faster lane than a vehicle focused on minimizing its emissions would choose, since faster speeds come with higher emissions. Thus, the action of choosing a faster lane taken to satisfy a travel time objective adversely affects the global emission levels and therefore works against the emissions reduction objective. The tendency of agents focused on reducing emissions to choose the slower lanes and the inclination of vehicles interested in reducing travel time to choose the faster lanes result in higher congestion levels in those lanes, and therefore work against the congestion reduction objective.

The preferred speed and objective emphasis values determine the target speed and target lane for each player. For TT players $TargetSpeed = PreferredSpeed + ObjectiveEmphasis$. For EL players $TargetSpeed = PreferredSpeed - ObjectiveEmphasis$. For CL players $TargetSpeed = PreferredSpeed$. For all players the target lane is the lane where the target speed falls within the allowed speed range.

The system maintains 9 atomic actions. Only single lane changes are allowed and the acceleration/deceleration factor is 1 (on a speed scale from 1 to 10).

The feasibility of separate lane and speed actions is, as before, derived by Algorithm 1. The availability of atomic lane/speed action combinations is shown in Algorithm 2.

**Algorithm 2:** Feasible Atomic Action Derivation

```
if TargeLane above CurrentLane and CanMoveUp then
    CanMoveUpAndMaintainSpeed = true;
    if CurrentSpeed < TargetSpeed then
        CanMoveUpAndAccelerate = true;
    else if CurrentSpeed > TargetSpeed then
        CanMoveUpAndDecelerate = true;
    else
        if CurrentSpeed < HighwayMaxSpeed then
            CanMoveUpAndAccelerate = true;
        if CurrentSpeed > LaneUpMinSpeed then
            CanMoveUpAndDecelerate = true;
if TargeLane below CurrentLane and CanMoveDown then
    CanMoveDownAndMaintainSpeed = true;
    if CurrentSpeed < TargetSpeed then
        CanMoveDownAndAccelerate = true;
    else if CurrentSpeed > TargetSpeed then
        CanMoveDownAndDecelerate = true;
    else
        if CurrentSpeed > HighwayMinSpeed then
            CanMoveDownAndDecelerate = true;
        if CurrentSpeed < LaneDownMaxSpeed then
            CanMoveUpAndAccelerate = true;
CanStayAndMaintainSpeed = true;
if (TargetLane = CurrentLane and AgentObjective != CL) or
 TargetLane != CurrentLane then
    if CurrentSpeed < LaneMaxSpeed and CanAccelerate then
        CanStayAndAccelerate = true;
    if CurrentSpeed > LaneMinSpeed and CanDecelerate then
        CanStayAndDecelerate = true;
if TargetLane = CurrentLane and AgentObjective = CL then
    if CurrentSpeed < TargetSpeed and CanAccelerate then
        CanStayAndAccelerate = true;
    else if CurrentSpeed > TargetSpeed and CanDecelerate then
        CanStayAndDecelerate = true;
    else
        Calculate RelativeLaneCongestion in distance ±x;
        /* value of x determined by mechanism */
        if CanMoveUp then
            Calculate RelativeLaneUpCongestion in distance ±x;
        if CanMoveDown then
            Calculate RelativeLaneDownCongestion in distance ±x;
        MinCongestion = min(valid RelativeLaneUpCongestion,
         RelativeLaneCongestion, RelativeLaneDownCongestion);
        /* break ties in favor of the higher lane */
        TargetLane = the lane with MinCongestion;
        TargetSpeed =
         TargetSpeed − ObjectiveEmphasis ∗ (1 − MinCongestion);
        if TargetSpeed < TargetLaneMinSpeed then
            TargetSpeed = TargetLaneMinSpeed;
        if TargetSpeed > TargetLaneMaxSpeed then
            TargetSpeed = TargetLaneMaxSpeed;
        if TargetLane = LaneUp then
            CanMoveUpAndAccelerate = true;
            CanMoveUpAndMaintainSpeed = true;
        else if TargetLane = LaneDown then
            CanMoveDownAndMaintainSpeed = true;
            CanMoveDownAndDecelerate = true;
        if CanAccelerate and CurSpeed != LaneMaxSpeed then
            CanStayAndAccelerate = true;
        if CanDecelerate and CurSpeed != LaneMinSpeed then
            CanStayAndDecelerate = true;
```

The actual ranking of feasible lane/speed actions is decided by rule sets for each objective taking into account the target lane and speed, the current speed, the current traffic lane and its speed restrictions, and the physical availability of space as determined by the neighboring agents. Travel Time (TT) and Emission Level (EL) vehicles strive to reach and maintain their target lane and speed as determined by the agents' relationships between preferred speed and objective emphasis). Congestion Level (CL) vehicles act to reach their target lane and speed and then consider and compare neighboring lanes' relative congestion levels. The least obstructed lane is selected as the new target lane and the level of objective emphasis is used to determine the new target speed (lines 2.40–2.46). All players rank action combinations depending on where their current speed setting and lane position are in relation to their targets.

These are the abstract concepts relevant to the test bed.

## 5. Simulation

The simulation of the presented mechanism for lane and speed allocation among driverless vehicles on a multi-lane highway was written, tested, and verified in NetLogo.

### 5.1. Settings

In the current implementation the maximum number of lanes is set to 5, but that upper bound was chosen to represent real world conditions and is not absolute. With small adjustments to scale up the legal value ranges of speed variables and objective emphasis the mechanism was successfully tested for a section of road with 11 lanes one way, to mimic California State route 22, considered to be one of the widest highways in the world. With appropriate small changes the model can be applied to roads with an arbitrary (but realistic) number of lanes.

In the simulation, for simplicity, agents are assigned the same physical attributes even though in the real world vehicle performance and structure characteristics influence and restrict the speed at which a vehicle can travel, the maneuvers it can perform, and the emission levels it produces. Adding size and performance attributes to the model is a minor extension.

Each round (cycle) of the game lasts for 10 time slices (ticks). 2 ticks are used for mechanism operations, while the remaining 8 are used for further forward progression, mechanism housekeeping, and as an adjustment period.

### 5.2. Happiness and patience

The simulation includes an indicator for *happiness*. "Happy" agents have reached their target lane and are travelling in it at their target speed. What happens afterwards however, differs according to the governing objective. Happy TT and EL vehicles strive to maintain these perceived ideal settings, only occasionally making overtake maneuvers where necessary, or making mechanism sanctioned changes if there is conflict present and they did not win their highest valued actions. For CL agents on the other hand, a happy state indicates that it is time to look to the neighboring lanes and see if it is possible to make a switch to a lane less congested than their current one. In order for CL agents to not inadvertently deviate too far from their original preferred speed due to traffic density circumstances, in the test bed their target speed and target lane are reset back to the initial values every 10 cycles.

The ability of agents to reach (and maintain) a happy state depends on overall traffic. Once in its target lane, an agent is more willing to make speed changes than to leave the lane. In order to avoid situations in which agents are stuck in an inferior condition (in the target lane but driving less than the target speed) for a long time, the simulation also includes the additional feature of *patience* for all agents. Patience reflects the sensitivity of agents to less than ideal conditions. For each time slice during which the vehicle is blocked by a slower driving vehicle directly ahead, it loses a unit of patience. In the time slice during which agents determine their feasible action valuations, if the patience level has dipped below zero the agent will attempt an overtake maneuver. The test bed implements right-hand traffic and allows for passing of vehicles both on the left (first choice) and on the right (only if passing on the left is not available at the moment). The pass on the right option (overtaking on the inside) is easily turned off if local traffic laws prohibit it. In the simulation, the nine atomic actions presented in Section 3.4 can be assigned a unique value between 0.1 and 0.9. When patience has run out, should a move up or down be physically possible, the *move up and accelerate* or *move down and accelerate* action is assigned a valuation of 1, signaling its precedence over all other possibilities. These actions are also coupled with a higher than usual acceleration factor to achieve the goal of passing the blocking vehicle(s) quickly while disrupting traffic as little as possible. For simplicity, the simulation does not currently evaluate the probability of successful completion of an overtake maneuver before undertaking it.

### 5.3. Resolving conflict

Once the mechanism manager receives all bids, it evaluates them for conflict. Each agent that has an uncontested top lane/speed action choice automatically receives approval for it, its inferior bids are deleted and the agent is excluded from further consideration during the round. AVs with only one available action also receive automatic approval. The bids of all other agents make up the conflict set, for which an allocation is to be constructed.

**Table 2**
Mechanism Design for Lane and Speed Allocation Among Autonomous Vehicles on a Highway Model - 100,000 Cycle Test Run Conflict-only Allocation Results.

| Traffic Density | Conflict Set Allocation Size | | | | Total Allocations | Payment Collected |
|---|---|---|---|---|---|---|
| | 2 agents | 3 agents | 4 agents | 5+ agents | | |
| 25% | 1467 | 21 | 4 | 0 | 1492 | 342.0 |
| 50% | 7392 | 264 | 277 | 19 | 7952 | 1842.5 |
| 75% | 8629 | 452 | 450 | 32 | 9563 | 2341.6 |

**Table 3**
Mechanism Design Model 100,000 Cycle Test Run Speed and Happiness Results.

| Performance Indicator | Traffic Density | | |
|---|---|---|---|
| | 25% | 50% | 75% |
| Average happiness of TT and EL agents | 88.2% | 70.6% | 66.0% |
| Minimum happiness of TT and EL agents | 50% | 11.1% | 5% |
| Maximum happiness of TT and EL agents | 100% | 100% | 92.5% |
| Time slices at 100% | 175,678 | 38 | - |
| Average happiness of all agents | 72.7% | 56.9% | 54.1% |
| Average difference of current to target speed for TT agents | −0.09 | −0.49 | −0.3 |
| Maximum over/under difference of average current speed to average target speed for TT agents | 1.25/2.63 | 0.61/3.57 | 0.65/3.8 |
| Average difference of current to target speed for EL agents | +0.02 | +0.02 | +0.1 |
| Maximum over/under difference of average current speed to average target speed for EL agents | 2/1.33 | 1.84/0.77 | 3.7/0.5 |
| Average difference of current to target speed for CL agents | +0.52 | +0.5 | +0.43 |
| Maximum over/under difference of average current speed to average target speed for CL agents | 3.5/1.92 | 2.65/1.42 | 1.85/1.65 |

During a cycle, conflict may be entirely absent or be present among any number of agents. When the set of incompatible bids is not empty the system also calculates cost. If faced with conflict, to improve efficiency, the simulation only produces the top allocation candidates. By sorting bids and continually truncating the search space the model guarantees to find the social utility maximizing allocation in shorter time.

### *5.4. Test runs*

The simulation was used for test runs of 1,000,000 time slices (100,000 full rounds) each at different levels of traffic density. The number of vehicles was evenly split between agents adhering to the 3 different objectives. The legal range of valuations of lane/speed actions and thus of price incurred was [0,1]. The size and number of constructed (conflict-only) allocations and the amount of payment collected for each test run are summarized in Table 2.

Table 3 lists some of the simulation results for the happiness and speed attributes. Due to the different nature of a happy state for CL agents, they were partially separated out. It is plain to see that as traffic volume increases, the ability of vehicles to maintain goal positions and velocities diminishes. At 25% traffic density all TT and EL agents were simultaneously happy 17.6% of the time as opposed to a mere 0.0038% when half the road was occupied by vehicles. As shown in Table 2, when traffic volume doubled, the level of conflict increased over 5 times as reflected by the total number of conflict-resolving allocations triggered by agents' wishes. Still, even at 75% traffic density, conflict that could not be resolved by preliminary bid reduction was present in just 9563 rounds out of 100,000 (9.6%) and in the vast majority of these cases the simulation successfully reduced the conflict set down to just two vehicles, which is trivial for the allocation function to resolve.

The diversity of speeds falling within the allowed range for a single lane results in some vehicles inadvertently blocking others behind them and slowing them down. Before they adjust their travel pace, AVs entering faster lanes at speeds temporarily at around or under the lane minimum speed can slow other lane occupants down as well. This resulted in Travel Time objective agents moving on average at a rate slightly under their goal. However, the average difference between current and target speeds for TT agents was still less than half a unit (on a 1–10 scale), so the mechanism did a good job of creating allocations benefiting the collective good that did not overly impede traffic. Emission Level agents travelled on average slightly above their target speed due to speeding up for overtake maneuvers they performed when blocked by slower vehicles for a period of time long enough to run their patience level down. By design CL agents varied their targets but the test run results showed that on average they too were able to travel very close to their goal. Even with restricted movement at 75% traffic density, the average happiness of agents was still high. Recall that a state of happiness was only indicated for vehicles currently travelling according to their *exact* wishes so agents who were only marginally removed from their happy state were not recorded. The mechanism as designed, achieved satisfactory outcomes for the whole group.

Fig. 4 shows a test run at the completion of the millionth time slice. Vertical lines denote just implemented lane changes, smiley face avatars denote vehicles in a happy state.

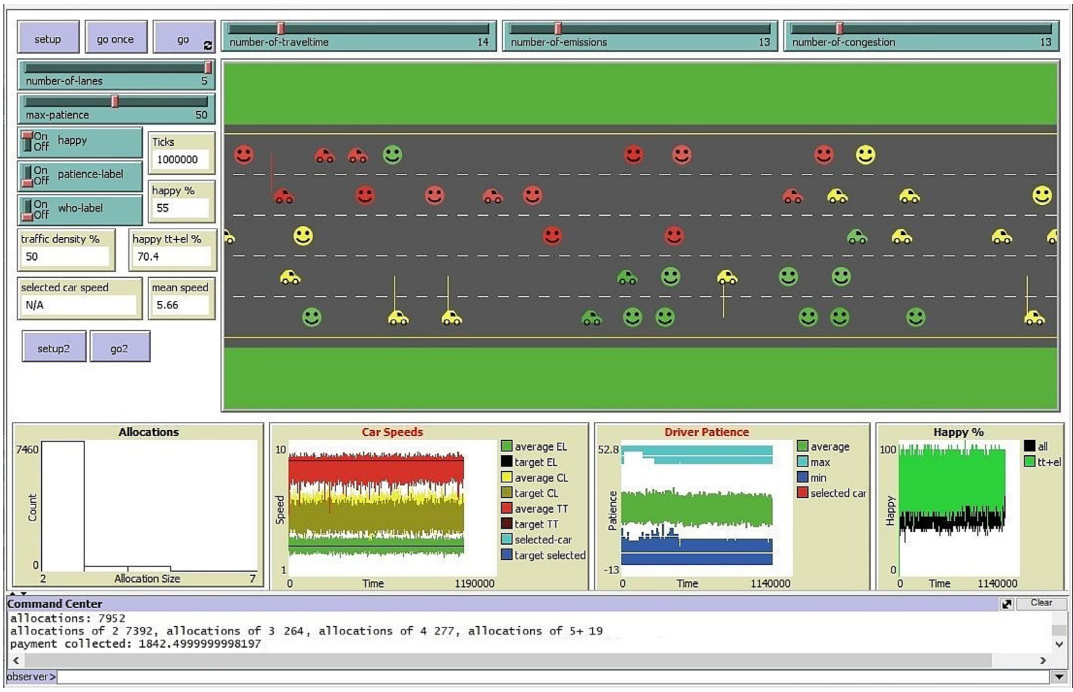


**Fig. 4.** NetLogo Simulation at a Completed 1000,000 Time Slice Test Run for Traffic Density of 50%.

### *5.5. Time complexity, optimization, and scalability*

In the simulation's current iteration, bid creation takes place simultaneously for all $n$ agents in runtime $O(1)$ for each agent. Each of the $n$ agents has at most $m$ bids, where $m$ is a constant. For the 2-dimensional speed/lane case discussed here, $m = 9$.

Agents insert their bids into the proper spatial order in a list for each of the $l$ lanes, where $l$ is a constant. This happens in $O(n)$ time with a conventional insertion strategy.

The current implementation of the mechanism's conflict evaluation and resolution functionality contains two steps:

1) *Bid reduction* - the bid set is reduced by approving only-option and uncontested top-option bids and removing any subpar bids for those agents,
2) *Conflict-free allocation extraction* - the remaining conflicted bids are arranged in the conflict-free configuration delivering the highest combined utility

**Lemma 2.** The mechanism runs in $O(n^3)$ time (worst case).

*Proof.* During step 1, bid reduction, the mechanism iterates through each of the $l$ lanes' bid lists, looking for two kinds of bids. For only-option bids, the algorithm looks to the bid's left and right and deletes any other claims within the required safety distance. For unconflicted best-option bids, the algorithm needs to consult at most 3 bids ahead and 3 bids behind to find out if there's conflict with others, including the same agent's own bids. Subpar choices for all agents with approved top-option bids are consecutively removed with a conventional search strategy.

These operations are currently implemented with nested loops. In the worst case, there are

$$\sum_{i=1}^{l}\left(\underbrace{\sum_{j=1}^{x}\left(\sum_{k=j-1}^{1}+\sum_{p=j+1}^{x}\right)}_{only-option\,bid\,reduction)}+\underbrace{\sum_{j=1}^{x}}_{top-option\,bid\,reduction}+\underbrace{\sum_{j=1}^{x}}_{remove\,subpar\,bids}\right)$$

iterations, where $x$ is the number of bids. The combined runtime is thus $O(n^2)$, since $l = const$ and at most $x = n * 3$.

Step 2, conflict-free allocation extraction, is only executed when after step 1 terminates not all agents have received approved actions. For the size of the conflict set $c$, the algorithm currently finds a conflict-free allocation in at most $\sum_{i=1}^{c}\sum_{j=1}^{9}\sum_{k=1}^{c}\sum_{o=1}^{9}\sum_{p=1}^{c-j-1}$ iterations, bringing the worst case runtime of the whole mechanism to $O(n^3)$. □

While $O(n^3)$ runtime sounds like bad news, numerical analysis from test runs of the mechanism (see Section 5.4) shows that the constraints of the environment and preliminary bid reduction strategies have an overwhelmingly positive effect on the computational burden placed on the allocation function. Enforcing lane speeds and an appropriate buffer zone between vehicles minimizes the number of initial bids deemed feasible and thus limits the size of the bid set. Bid ordering ensures that the only-option and top-option bid reduction operations usually terminate early. The test runs show what the bid reduction strategy delivers a conflict free set over 90% of the time, so the computationally heavy allocation function does not have to be executed at all. And because of local partitioning, the size of the conflict set has a low upper bound and is minimal or trivial in the vast majority of cases.

There are several potential avenues for runtime reduction and optimization. When the mechanism governs a long stretch of road, the conflict set can be split into disjoint local clusters, further reducing its size. The local conflict subsets can be processed simultaneously to save time. As a matter of fact, after receiving the agents' bids, the manager can run a partitioning algorithm to subdivide its area of influence into smaller sets without overlapping bids to process separately. The allocation function can be implemented to recognize duplicate allocations and ignore them, as well as to abandon suboptimal allocations early. Depending on traffic density, the mechanism manager can set an upper bound on the number of agents making up the conflict set. Should there be too much conflict, the manager can skip the allocation altogether, assigning each conflicted agent the atomic action to stay in the same lane and maintain the current speed. Several seconds of forward progression will inevitably result in a different configuration at the beginning of the next round. This work focuses on the mechanism as a concept and does not further investigate its optimization.

As outlined, many factors contribute to conflict reduction on a scale sufficient enough to only necessitate firing the computationally expensive allocation function in a small number of rounds. Vehicles regulate their own speed so if an agent has to slow down because of a lane change made some distance ahead, it can do that without having to bid for it. Speed and safety distance constraints further cut down the set of feasible actions. In lighter traffic, there are fewer total bids and fewer conflicts. Denser traffic results in fewer feasible bids. Lane changes only affect close neighbors so the same manager can divide its conflict set into multiple smaller local clusters and manage them separately. The preliminary bid reduction strategy cuts down the size of the conflict set, in most cases eliminating it entirely. Should the allocation function have to be executed, it is only run on the partitioned conflict set(s). The option to skip allocation when too much conflict is present and the substantially diminished size of the local conflict subset greatly reduce the calculation burden and alleviate scalability concerns connected to the Clarke pivot rule.

### *5.6. Additional features and future extensions*

Additional features specific to the application domain can be added to the simulation while still keeping the integrity of the underlying model. For added behavioral flexibility, agents could be allowed to switch between objectives at will. Auxiliary objectives can be modelled and included to represent AV intentions - e.g., maximizing fuel efficiency, following a particular lead vehicle, etc. For progress improvement, agents could sense the feasibility of overtake maneuvers before undertaking them. The current happiness metric can be extended by degrees of satisfaction. The patience metric and the circumstances that cause it to decrease could be refined. The payment scale can be changed and monetary or other incentives can be added to the resolution of utility ties. The allocation algorithm can be augmented with degrees of externality to reflect the severity of imposition an approved action for an agent would inflict on others. The resulting scale can be used to make more sophisticated allocation decisions or can result in the utilization of a multiplier for mechanism payment according to the circumstances. The simulation should be equipped with attributes allowing for the diversity of agents in terms of physical features. Those features in combination with current speeds should be taken into account when calculating safe distances during travel. Next, inclusions can be made to add obstacles and special vehicles that can be treated as unavailable road positions. If separate mechanisms to enable the utilization of highway on and off ramps are added to the model, the mechanism can be fine-tuned to a potential real world deployment.

## 6. Conclusion

This work presented a game-theoretic multi-objective model for the allocation of lane positions and speed settings among autonomous vehicles on a multi-lane highway. Rooted in mechanism design, the approach delivers a social utility maximizing strategyproof mechanism for negotiation and conflict resolution. The localized nature of lane-changes ensures the scalability of the model. The system also benefits from performance advantages granted to it by its deployment at the edge of the Internet of Things.

A prototyped NetLogo implementation was presented equipped with a limited set of shared objectives governing patterns of driving. The empirical model includes rule sets for determining action feasibility and for the ranking of eventualities. It demonstrates the ability of the system to create allocations that resolve conflict in favor of the collective good. Features of the system are adjustable to fit the deployment scenario.

Even though parts of the presented model are specific to the application domain, by making the context-appropriate changes, the approach can be implemented and utilized in any field with similar demands on its resources. The proposed mechanism heralds a step toward autonomous negotiation among machines and it equips interconnected smart cyber-physical systems with efficient and meaningful ways to privately communicate preferences to an arbitrating body. The model requires minimal resources and gives agents the ability to take part in a negotiation process free from manipulation and resulting in a socially satisfactory outcome.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

[1] S. Severi, F. Sottile, G. Abreu, C. Pastrone, M. Spirito, F. Berens, M2m technologies: enablers for a pervasive internet of things, in: Proceedings of the European Conference on Networks and Communications (EuCNC), 2014, pp. 1–5, doi:10.1109/EuCNC.2014.6882661.
[2] F. Mattern, C. Floerkemeier, From Active Data Management to Event-Based Systems and More: Papers in Honor of Alejandro Buchmann On the Occasion of His 60th Birthday, Springer Berlin Heidelberg, 2010, doi:10.1007/978-3-642-17226-715.
[3] E.A. Lee, Cyber-physical systems - are computing foundations adequate? Position Paper for NSF Workshop On Cyber-Physical Systems: Research Motivation, Techniques and Roadmap, 2006 URL http://chess.eecs.berkeley.edu/pubs/329.html.
[4] G. Fortino, A. Guerrieri, W. Russo, C. Savaglio, Integration of agent- based and cloud computing for the smart objects-oriented iot, in: Proceedings of the 2014 IEEE 18th International Conference on Computer Supported Cooperative Work in Design (CSCWD), 2014, pp. 493–498, doi:10.1109/CSCWD.2014.6846894.
[5] H. Yu, Z. Shen, C. Leung, From internet of things to internet of agents, in: Proceedings of the IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, 2013, pp. 1054–1057, doi:10.1109/GreenComiThings-CPSCom.2013.179.
[6] M. Satyanarayanan, The emergence of edge computing, Computer 50 (1) (2017) 30–39, doi:10.1109/MC.2017.9.
[7] P.G. Lopez, A. Montresor, D. Epema, A. Datta, T. Higashino, A. Iamnitchi, M. Barcellos, P. Felber, E. Riviere, Edge-centric computing: vision and challenges, SIGCOMM Comput. Commun. Rev. 45 (5) (2015) 37–42, doi:10.1145/2831347.2831354.
[8] J. Ren, Y. Pan, A. Goscinski, R.A. Beyah, Edge computing for the internet of things, IEEE Netw. 32 (1) (2018) 6–7, doi:10.1109/MNET.2018.8270624.
[9] N.J. Goodall, Can you program ethics into a self-driving car? IEEE Spectr. 53 (6) (2016) 28–58, doi:10.1109/MSPEC.2016.7473149.
[10] V. Skrickij, E. S˘abanovi˘c, V. Z˘uraulis, Autonomous road vehicles: recent issues and expectations, IET Intell. Transp. Syst. 14 (6) (2020) 471–479, doi:10.1049/iet-its.2018.5513.
[11] K.W. Schmidt, Cooperative adaptive cruise control for vehicle following during lane changes, IFAC-PapersOnLine 50 (1) (2017) 12582–12587 (TUBITAK) [Award 115E372] 20th IFAC World Congress, doi:10.1016/j.ifacol.2017.08.2199.
[12] M. Kamali, L.A. Dennis, O. McAree, M. Fisher, S.M. Veres, Formal verification of autonomous vehicle platooning, Sci. Comput. Program. 148 (2017) 88–106 special issue on Automated Verification of Critical Systems (AVoCS 2015), doi:10.1016/j.scico.2017.05.006.
[13] Y. Ali, Z. Zheng, M.M. Haque, M. Wang, A game theory-based approach for modelling mandatory lane-changing behaviour in a connected environment, Transp. Res. Part C Emerg. Technol. 106 (2019) 220–242, doi:10.1016/j.trc.2019.07.011.
[14] L. Claussmann, M. Revilloud, D. Gruyer, S. Glaser, A review of motion planning for highway autonomous driving, IEEE Trans. Intell. Transp. Syst. 21 (5) (2020) 1826–1848, doi:10.1109/TITS.2019.2913998.
[15] A.A. Joshi, D.L. Peters, J.M. Bastiaan, Autonomous lane change control using proportional-integral-derivative controller and bicycle model, in: Proceedings of the WCX SAE World Congress Experience, SAE International, 2020, doi:10.4271/2020-01-0215.
[16] S. Dixit, S. Fallah, U. Montanaro, M. Dianati, A. Stevens, F. Mccullough, A. Mouzakitis, Trajectory planning and tracking for autonomous overtaking: state-of-the-art and future prospects, Annu. Rev. Control 45 (2018) 76–86, doi:10.1016/j.arcontrol.2018.02.001.
[17] S.V. Dorff, B. B¨oddeker, M. Kneissl, M. Fr¨anzle, A fail-safe architecture for automated driving, in: Proceedings of the Design, Automation Test in Europe Conference Exhibition (DATE), 2020, pp. 828–833, doi:10.23919/DATE48585.2020.9116283.
[18] D. Bevly, X. Cao, M. Gordon, G. Ozbilgin, D. Kari, B. Nelson, J. Woodruff, M. Barth, C. Murray, A. Kurt, K. Redmill, U. Ozguner, Lane change and merge maneuvers for connected and automated vehicles: a survey, IEEE Trans. Intell. Veh. 1 (1) (2016) 105–120, doi:10.1109/TIV.2015.2503342.
[19] Y. Ding, W. Zhuang, Y. Qian, H. Zhong, Trajectory planning for automated lane-change on a curved road for collision avoidance, in: Proceedings of the WCX SAE World Congress Experience, SAE International, 2019, doi:10.4271/2019-01-0673.
[20] M. Wang, Z. Wang, S. Paudel, M. Schwager, Safe distributed lane change maneuvers for multiple autonomous vehicles using buff input cells, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2018, pp. 4678–4684, doi:10.1109/ICRA.2018.8460898.
[21] H. Li, Y. Luo, Study on steering angle input during the automated lane change of electric vehicle, in: Proceedings of the Intelligent and Connected Vehicles Symposium, SAE International, 2017, doi:10.4271/2017-01-1962.
[22] M.L. Ho, P.T. Chan, A.B. Rad, Lane change algorithm for autonomous vehicles via virtual curvature method, J. Adv. Transp. 43 (1) (2009) 47–70, doi:10.1002/atr.5670430104.
[23] J. Laval, C. Daganzo, Lane-changing in traffic stream, Transp. Res. Part B Methodol. 40B (2006) 251–264, doi:10.1016/j.trb.2005.04.003.
[24] J. Nilsson, M. Brannstrom, E. Coelingh, J. Fredriksson, Lane change maneuvers for automated vehicles, IEEE Trans. Intell. Transp. Syst. 18 (5) (2017) 1087–1096, doi:10.1109/TITS.2016.2597966.
[25] C. Katrakazas, M. Quddus, W.-.H. Chen, L. Deka, Real-time motion planning methods for autonomous on-road driving: state-of-the-art and future research directions, Transp. Res. Part C Emerg. Technol. 60 (2015) 416–442, doi:10.1016/j.trc.2015.09.011.
[26] L. Guo, P.-.S. Ge, M. Yue, Y.-.B. Zhao, Lane changing trajectory planning and tracking controller design for intelligent vehicle running on curved road, Math. Probl. Eng. (2014) 2014, doi:10.1155/2014/478573.
[27] C. Pek, P. Zahn, M. Althoff, Verifying the safety of lane change maneuvers of self-driving vehicles based on formalized traffic rules, in: Proceedings of the IEEE Intelligent Vehicles Symposium (IV), 2017, pp. 1477–1483, doi:10.1109/IVS.2017.7995918.
[28] M. Naumann, H. Ko¨nigshof, C. Stiller, Provably safe and smooth lane changes in mixed traffic, in: Proceedings of the IEEE Intelligent Transportation Systems Conference (ITSC), 2019, pp. 1832–1837, doi:10.1109/ITSC.2019.8917461.
[29] M. Hilscher, S. Linker, E.-.R. Olderog, A.P. Ravn, An abstract model for proving safety of multi-lane traffic manoeuvres, in: S. Qin, Z. Qiu (Eds.), Formal Methods and Software Engineering, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 404–419, doi:10.1007/978-3-642-24559-628.
[30] B. Qiao, X. Wu, Lane change control of autonomous vehicle with real-time rerouting function, in: Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), 2019, pp. 1317–1322, doi:10.1109/AIM.2019.8868634.
[31] C. Wang, Q. Sun, Z. Li, H. Zhang, Human-like lane change decision model for autonomous vehicles that considers the risk perception of drivers in mixed traffic, Sensors 20 (2020) 2259, doi:10.3390/s20082259.
[32] M. Atagoziyev, K.W. Schmidt, E.G. Schmidt, Lane change scheduling for autonomous vehicles**this work was supported by the scientific and technological research council of Turkey (tubitak) [award 115e372], IFAC-PapersOnLine 49 (3) (2016) 61–66, doi:10.1016/j.ifacol.2016.07.011.
[33] Y. Luo, G. Yang, M. Xu, Z. Qin, K. Li, Cooperative lane-change maneuver for multiple automated vehicles on a highway, Autom. Innov. 2 (3) (2019) 157–168, doi:10.1007/s42154-019-00073-1.
[34] T. Dao, C.M. Clark, J.P. Huissoon, Optimized lane assignment using inter-vehicle communication, in: Proceedings of the IEEE Intelligent Vehicles Symposium, 2007, pp. 1217–1222, doi:10.1109/IVS.2007.4290284.
[35] A.P. Chouhan, G. Banda, K. Jothibasu, A cooperative algorithm for lane sorting of autonomous vehicles, IEEE Access 8 (2020) 88759–88768, doi:10.1109/ACCESS.2020.2993200.
[36] H. Bai, J. Shen, L. Wei, Z. Feng, Accelerated lane-changing trajectory planning of automated vehicles with vehicle-to-vehicle collaboration, J. Adv. Transp. (2017) 2017, doi:10.1155/2017/8132769.

[37] D. Wang, M. Hu, Y. Wang, J. Wang, H. Qin, Y. Bian, Model predictive control–based cooperative lane change strategy for improving traffic flow, Adv. Mech. Eng. 8 (2) (2016) 1687814016632992, doi:10.1177/1687814016632992.
[38] J. Nie, J. Zhang, W. Ding, X. Wan, X. Chen, B. Ran, Decentralized cooperative lane-changing decision-making for connected autonomous vehicles*, IEEE Access 4 (2016) 9413–9420, doi:10.1109/ACCESS.2017.2649567.
[39] J. Ni, J. Han, F. Dong, Multivehicle cooperative lane change control strategy for intelligent connected vehicle, J. Adv. Transp. (2020) 2020, doi:10.1155/2020/8672928.
[40] T. Li, J. Wu, C. Chan, M. Liu, C. Zhu, W. Lu, K. Hu, A cooperative lane change model for connected and automated vehicles, IEEE Access 8 (2020) 54940–54951, doi:10.1109/ACCESS.2020.2981169.
[41] A. Ji, D. Levinson, A review of game theory models of lane changing, Transp. A Transp. Sci. 16 (3) (2020) 1628–1647, doi:10.1080/23249935.2020.1770368.
[42] A. Talebpour, H.S. Mahmassani, S.H. Hamdar, Modeling lane-changing behavior in a connected environment: a game theory approach, Transp. Res. Part C Emerg. Technol. 59 (2015) 216–232 special Issue on International Symposium on Transportation and Traffic Theory, doi:10.1016/j.trc.2015.07.007.
[43] M. Wang, S.P. Hoogendoorn, W. Daamen, B. van Arem, R. Happee, Game theoretic approach for predictive lane-changing and car-following control, Transp. Res. Part C Emerg. Technol. 58 (2015) 73–92, doi:10.1016/j.trc.2015.07.009.
[44] H. Yu, H.E. Tseng, R. Langari, A human-like game theory- based controller for automatic lane changing, Transp. Res. Part C Emerg. Technol. 88 (2018) 140–158, doi:10.1016/j.trc.2018.01.016.
[45] E. Lovellette, H. Hexmoor, K. Rodriguez, Automated argumentation for collaboration among cyber-physical system actors at the edge of the internet of things, Internet of Things 5 (2019) 84–96, doi:10.1016/j.iot.2018.12.002.
[46] D. Lin, L. Li, S.E. Jabari, Pay to change lanes: a cooperative lane-changing strategy for connected/automated driving, Transp. Res. Part C Emerg. Technol. 105 (2019) 550–564, doi:10.1016/j.trc.2019.06.006.
[47] D. Fudenberg, J. Tirole, Game Theory, MIT Press, Cambridge, MA, 1991.
[48] J. Hartline, Mechanism Design and Approximation, 2017 http://jasonhartline.com/MDnA.
[49] N. Nisan, T. Roughgarden, E. Tardos, V. Vazirani, Algorithmic Game Theory, Cambridge University Press, 2007, doi:10.1017/CBO9780511800481.
[50] E. Maskin, Nash equilibrium and mechanism design, Games Econ. Behav. 71 (2011) 9–11.
[51] R.B. Myerson, Perspectives on mechanism design in economic theory, Am. Econ. Rev. 98 (3) (2008) 586–603, doi:10.2307/29730088.
[52] N. Nisan, in: Introduction to Mechanism Design (for Computer Scientists), Cambridge University Press, 2007, pp. 209–242, doi:10.1017/CBO9780511800481.011.
[53] W.S. Vickrey, Congestion theory and transport investment, Am. Econ. Rev. 59 (2) (1969) 251–260 URL http://www.jstor.org/stable/1823678.
[54] S.J. Rassenti, V.L. Smith, R.L. Bulfin, A combinatorial auction mech- anism for airport time slot allocation, Bell J. Econ. 13 (2) (1982) 402–417, doi:10.1017/9781316471609.019.
[55] J.M. Viegas, Making urban road pricing acceptable and effective: searching for quality and equity in urban mobility, Transp. Policy 8 (4) (2001) 289–294, doi:10.1016/S0967-070X(01)00024-5.
[56] T. Vanoutrive, Don't think of them as roads. Think of them as road transport markets: congestion pricing as a neoliberal political project, Prog. Plann. 117 (2017) 1–21, doi:10.1016/j.progress.2016.04.001.
[57] D. Wu, Y. Yin, S. Lawphongpanich, H. Yang, Design of more equitable congestion pricing and tradable credit schemes for multimodal transportation networks, Transp. Res. Part B Methodol. 46 (9) (2012) 1273–1287, doi:10.1016/j.trb.2012.05.004.
[58] Y. Hara, E. Hato, A car sharing auction with temporal-spatial od connection conditions, Transp. Res. Proc. 23 (2017) 22–40 papers Selected for the 22nd International Symposium on Transportation and Traffic Theory Chicago, Illinois, USA, 24-26 July 2017, doi:10.1016/j.trpro.2017.05.003.
[59] Y. Hara, E. Hato, Analysis of dynamic decision-making in a bicycle-sharing auction using a dynamic discrete choice model, Transportation 46 (1) (2019) 147–173, doi:10.1007/s11116-017-9795-x.
[60] K. Wada, T. Akamatsu, A hybrid implementation mechanism of trad- able network permits system which obviates path enumeration: an auction mechanism with day-to-day capacity control, Transp. Res. Part E Logist. Transp. Rev. 60 (2013) 94–112, doi:10.1016/j.tre.2013.05.008.
[61] M. Osawa, H. Fu, T. Akamatsu, First-best dynamic assignment of commuters with endogenous heterogeneities in a corridor network, Transp. Res. Part B Methodol. 117 (2018) 811–831 tRB:ISTTT-22, doi:10.1016/j.trb.2017.09.003.
[62] L. Benmohamed, S.M. Meerkov, Feedback control of highway congestion by a fair on-ramp metering, in: Proceedings of 1994 33rd IEEE Conference on Decision and Control, Vol. 3, 1994, pp. 2437–2442, doi:10.1109/CDC.1994.411448.
[63] Q. Tian, H.-J. Huang, H. Yang, Z. Gao, Efficiency and equity of ramp control and capacity allocation mechanisms in a freeway corridor, Transp. Res. Part C Emerg. Technol. 20 (1) (2012) 126–143 Special issue on Optimization in Public Transport+ISTT2011, doi:10.1016/j.trc.2011.05.005.
[64] Y. Han, A. Hegyi, Y. Yuan, S. Hoogendoorn, Validation of an ex- tended discrete first-order model with variable speed limits, Transp. Res. Part C Emerg. Technol. 83 (2017) 1–17, doi:10.1016/j.trc.2017.07.009.
[65] B. Zou, N. Kafle, O. Wolfson, J.J. Lin, A mechanism design based approach to solving parking slot assignment in the information era, Transp. Res. Part B Methodol. 81 (2015) 631–653 Optimization of Urban Transportation Service Networks, doi:10.1016/j.trb.2015.05.015.
[66] L. Han, D.Z. Wang, C. Zhu, The discrete-time second-best dynamic road pricing scheme, Transp. Res. Proc. 23 (2017) 322–340 Papers Selected for the 22nd International Symposium on Transportation and Traffic Theory Chicago, Illinois, USA, 24-26 July 2017, doi:10.1016/j.trpro.2017.05.019.
[67] G. Cantelmo, F. Viti, E. Cipriani, M. Nigro, A utility-based dynamic demand estimation model that explicitly accounts for activity scheduling and duration, Transp. Res. Proc. 23 (2017) 440–459 Papers Selected for the 22nd International Symposium on Transportation and Traffic Theory Chicago, Illinois, USA, 24-26 July 2017, doi:10.1016/j.trpro.2017.05.025.
[68] M. Lafkihi, S. Pan, E. Ballot, Rule-based incentive mechanism design for a decentralised collaborative transport network, Int. J. Prod. Res. 0 (0) (2019) 1–17, doi:10.1080/00207543.2019.1693658.
[69] M. Lai, X. Cai, in: Algorithmic Mechanism Design For Collaboration in Large-Scale Transportation Networks, Springer International Publishing, 2019, pp. 257–282, doi:10.1007/978-3-030-22788-39.
[70] Y. Hara, Behavioral mechanism design for transportation services: laboratory experiments and preference elicitation cost, Transp. Res. Part B Methodol. 115 (2018) 231–245, doi:10.1016/j.trb.2018.07.008.
[71] C. Cheng, Mechanism design for enterprise transportation outsourcing based on combinatorial auction, in: Proceedings of the 11th International Conference on Service Systems and Service Management (ICSSSM), 2014, pp. 1–5, doi:10.1109/ICSSSM.2014.6874140.
[72] J.C. Harsanyi, Games with incomplete information played by "bayesian" players, i-iii. Part ii. Bayesian equilibrium points, Manag. Sci. 14 (5) (1968) 320–334 URL http://www.jstor.org/stable/2628673.
[73] D. Antos, A. Pfeffer, Representing Bayesian games without a common prior, in: Proceedings of the 9th International Conference On Autonomous Agents and Multiagent Systems: Volume 1 - Volume 1, AA- MAS '10, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2010, pp. 1457–1458.
[74] M. Hoeing, P. Dasgupta, P. Petrov, S. O'Hara, Auction-based multi- robot task allocation in comstar, in: Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS '07, ACM, New York, NY, USA, 2007, pp. 1–8, doi:10.1145/1329125.1329462.
[75] P.B. Sujit, R. Beard, Distributed sequential auctions for multiple uav task allocation, in: Proceedings of the American Control Conference, 2007, pp. 3955–3960, doi:10.1109/ACC.2007.4282558.
[76] D. Bertsekas, The auction algorithm for assignment and other network flow problems: a tutorial, Interfaces 20 (4) (1990) 133–149, doi:10.1287/inte.20.4.133.
[77] E.H. Clarke, Multipart pricing of public goods, Public Choice 11 (1) (1971) 17–33, doi:10.1007/BF01726210.
[78] W. Vickrey, Counterspeculation, auctions, and competitive sealed tenders, J. Financ. 16 (1) (1961) 8–37, doi:10.1111/j.1540-6261.1961.tb02789.x.
[79] T. Groves, Incentives in teams, Econometrica 41 (4) (1973) 617–631, doi:10.2307/1914085.