

Negotiation of Drivable Areas of Cooperative Vehicles for Conflict Resolution

Stefanie Manzinger and Matthias Althoff

Department of Informatics

Technische Universität München

Boltzmannstraße 3, 85748 Garching bei München, Germany

Email: {stefanie.manzinger, althoff}@in.tum.de

Abstract—We address the problem of cooperative conflict resolution for multi-vehicle motion planning in mixed-traffic scenarios, where automated and manually-driven vehicles co-exist. We propose a novel solution based on reachability analysis, which provides the drivable area of each collaborative traffic participant. Overlapping drivable areas are redistributed so that each traffic participant receives an individual area for motion planning. We do not stipulate a specific method for predicting the future motion of non-communicating traffic participants. Furthermore, uncertainties in the initial states of the cooperative vehicles, e.g. due to sensor noise, can be easily integrated. A byproduct of our approach is that collaborative groups can be automatically found by identifying conflicting drivable areas; if no conflict exists, collaboration becomes unnecessary. We demonstrate the redistribution of drivable areas with two numerical examples.

I. INTRODUCTION

Collaborative motion planning of various automated road vehicles is clearly superior in terms of achievable safety and comfort compared to computing individual motion plans. This is because individual motion planning is a special case of collaborative planning when vehicles are not communicating. Many promising approaches for multi-vehicle motion planning have been developed; however, dealing with mixed-traffic situations and uncertainty is still an open research topic. We propose a unified approach for cooperative conflict resolution based on the computation of drivable areas where automated and manually-driven vehicles share the road. We first review literature concerning specific applications like intersection management and merging; after, we discuss priority-based, market-based, and reservation-based approaches.

Much work on cooperative motion planning has been devoted to road intersections, since these are hotspots for traffic accidents. Collision avoidance at intersections using V2V-communication for cooperation is investigated in [1] under the consideration of model uncertainty and communication delays. Colombo et al. [2] solve scheduling problems to ensure safety during intersection passages.

Another line of research is the design of cooperative lane-changing and merging strategies. In [3], it is discussed how V2V-communication can be utilized for cooperative decision making: a distributed receding horizon control framework is set up to solve tasks of platooning and cooperative merging. Further lane-changing and merging control algorithms for platoons of vehicles are developed in [4], [5].

Frese et al. [6] exploit priority-based motion planning to decouple the multi-vehicle motion planning problem, such that trajectory planning can only be conducted for single vehicles. This decreases the computational complexity; however, the solution space is reduced. Bekris et al. [7] combine a sampling-based motion planner with a priority-based coordination scheme, which is compared with a message-passing protocol for distributed constraint optimization. Moreover, priority-based algorithms for intersection management and traffic flow control are elaborated in [8]–[10].

Recently, market-based approaches have received major interest for multi-vehicle coordination, since they allow the incorporation of individual as well as global objectives, making it useful to balance self-interested and collective goals. In [11], maneuver plans are negotiated and refined via model predictive control. Auction-based coordination strategies for intersections can be found in [12]–[14].

Finally, we review literature concerning reservation-based algorithms [15]–[18], where communicating vehicles reserve some sort of space-time slots by requesting them via a supervisor. It must be guaranteed that the space-time slots are not occupied by more than one vehicle in order to ensure safety. In [15]–[17], reservation-based algorithms are applied to intersections; [15] in particular divides the intersection into tiles which can be allocated to communicating vehicles. Marinescu et al. [18] implement a slot-based approach for the merging of on-ramp traffic. They propose combining the hierarchical approach of exploiting vehicle-to-infrastructure communication with the decentralized approach of utilizing inter-vehicle communication for vehicle coordination.

We propose an algorithm related to the idea of reservation-based algorithms. In contrast to previous work, we distribute drivable areas of cooperative vehicles such that overlapping drivable areas are unambiguously reallocated to single cooperative vehicles. The computation of the drivable areas is based on reachability analysis, which allows us to not only compute a set of drivable positions but also to determine the maximum velocity range to reach a distinct set of positions. Thus, we are not restricted to distributing complete road segments among the collaborative vehicles, but we can precisely resolve conflicts. Our approach reduces the search space when planning coordinated maneuvers for multiple vehicles, since the trajectory of a single vehicle is restricted to its associated

drivable area. Thus, the computational complexity of multi-vehicle motion planning can be decreased. Our method is suitable for identifying collaborative groups of vehicles for which motion planning can be conducted jointly. Moreover, it is possible to detect if collaboration with a cooperative vehicle becomes unnecessary. Since our method is set-based, the consideration of uncertainty in the initial position or velocity of the cooperative vehicles, e.g. due to sensor noise, is automatically supported. Our approach is applicable to mixed-traffic scenarios, where automated vehicles and human drivers co-exist.

The remainder of this paper is organized as follows: Sec. II introduces the problem statement and Sec. III presents the basic idea. A comprehensive description of our applied methods and our proposed algorithm is provided in Sec. IV. Sec. V demonstrates our approach on two numerical examples, followed by the conclusion in Sec. VI.

II. PROBLEM STATEMENT

Let us introduce \square as the placeholder for a variable and \square_n to denote the value of the corresponding variable of the n -th cooperative vehicle, $n \in \mathcal{N} := \{1, 2, \dots, N\}$. The system dynamics of the n -th vehicle is described by the differential equation

$$\dot{\mathbf{x}}_n(t) = f_n(\mathbf{x}_n(t), \mathbf{u}_n(t)),$$

where \mathbf{x}_n is the state, \mathbf{u}_n is the input, and t is the time. All possible initial states and the admissible inputs are bounded by sets: $\mathbf{x}_n(0) \in \mathcal{X}_{n,0}, \forall t: \mathbf{u}_n(t) \in \mathcal{U}_n$. We further introduce the solution of the differential equation $\dot{\mathbf{x}}_n(t) = f_n(\mathbf{x}_n(t), \mathbf{u}_n(t))$ as $\chi_n(t; \mathbf{x}_n(0), \mathbf{u}_n(\cdot))$, where $\mathbf{u}_n(\cdot)$ refers to the input trajectory.

The reachable set of the system $\dot{\mathbf{x}}_n(t) = f_n(\mathbf{x}_n(t), \mathbf{u}_n(t))$ is usually defined as the set of all states which can be reached from an initial set $\mathcal{X}_{n,0}$ at time t . However, the vehicles generally move in a structured environment cluttered with (time-dependent) obstacles represented by the set $\mathcal{O}(t) \subseteq \mathbb{R}^2$. Since we require the absence of collisions, each vehicle must not enter the set of forbidden states

$$\mathcal{F}_n(t) := \{\mathbf{x}_n(t) \in \mathcal{X}_n \mid \mathcal{Q}_n(\mathbf{x}_n(t)) \cap \mathcal{O}(t) \neq \emptyset\},$$

where $\mathcal{Q}_n(\mathbf{x}_n(t)) \subseteq \mathbb{R}^2$ denotes the occupancy of the n -th cooperative vehicle. Therefore, we restrict the reachable set $\mathcal{R}_n(\mathcal{X}_{n,0}, t)$ of the n -th vehicle to the set of states that can be reached without any collision with the obstacle set $\mathcal{O}(t)$:

$$\mathcal{R}_n(\mathcal{X}_{n,0}, t) := \left\{ \chi_n(t; \mathbf{x}_n(0), \mathbf{u}_n(\cdot)) \mid \mathbf{x}_n(0) \in \mathcal{X}_{n,0}, \right. \\ \left. \forall \tau \in [0, t]: \mathbf{u}_n(\tau) \in \mathcal{U}_n, \chi_n(\tau; \mathbf{x}_n(0), \mathbf{u}_n(\cdot)) \notin \mathcal{F}_n(\tau) \right\}.$$

The drivable area $\mathcal{D}_n(\mathcal{X}_{n,0}, t)$ of the n -th vehicle is given by the reachable positions at time t . We introduce the projection operator $\text{proj}(\cdot)$ to project a set of states to the position domain.

Definition 1 (Projection): Given that $\mathbf{x}(t) \in \mathcal{X}'$ contains the position $s_x(t)$ and $s_y(t)$ in x - and y -direction, we define the

mapping from a set of states $\mathcal{X}' \subseteq \mathcal{X}$ to the set of positions as

$$\text{proj}(\mathcal{X}') := \{[s_x(t), s_y(t)]^T \in \mathbb{R}^2 \mid \mathbf{x}(t) \in \mathcal{X}'\}.$$

Definition 2 (Drivable Area): The drivable area $\mathcal{D}_n(\mathcal{X}_{n,0}, t)$ of the n -th vehicle is defined as the projection of its reachable set $\mathcal{R}_n(\mathcal{X}_{n,0}, t)$: $\mathcal{D}_n(\mathcal{X}_{n,0}, t) := \text{proj}(\mathcal{R}_n(\mathcal{X}_{n,0}, t))$.

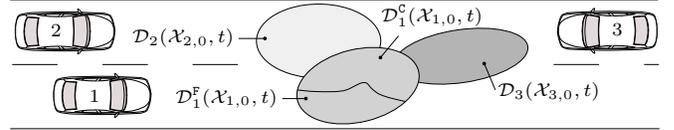


Fig. 1: Conflicting and conflict-free drivable area $\mathcal{D}_1^C(\mathcal{X}_{1,0}, t)$ and $\mathcal{D}_1^F(\mathcal{X}_{1,0}, t)$ of vehicle 1.

The collaborative vehicles do not only have to avoid the set of forbidden states $\mathcal{F}_n(t)$, but they also have to prevent accidents among each other. Given the drivable areas $\mathcal{D}_n(\mathcal{X}_{n,0}, t)$ of N collaborative vehicles at time t , the occupancies of the vehicles may overlap (see Fig. 1). The overlapping region defines the area where conflicts potentially arise and a collision may occur. Thus, the drivable area $\mathcal{D}_n(\mathcal{X}_{n,0}, t)$ can be partitioned into a conflicting and a non-conflicting region.

Definition 3 (Conflicting Drivable Area): We introduce the conflicting reachable set $\mathcal{R}_n^C(\mathcal{X}_{n,0}, t) \subseteq \mathcal{R}_n(\mathcal{X}_{n,0}, t)$ as the set of states $\mathbf{x}_n(t) \in \mathcal{R}_n(\mathcal{X}_{n,0}, t)$, where the occupancy $\mathcal{Q}_n(\mathbf{x}_n(t))$ of the n -th vehicle potentially intersects with the occupancy $\mathcal{Q}_k(\mathbf{x}_k(t))$ of another cooperative vehicle $k \in \mathcal{N} \setminus \{n\}$:

$$\mathcal{R}_n^C(\mathcal{X}_{n,0}, t) := \left\{ \mathbf{x}_n(t) \in \mathcal{R}_n(\mathcal{X}_{n,0}, t) \mid \mathbf{x}_k(t) \in \mathcal{R}_k(\mathcal{X}_{k,0}, t), \right. \\ \left. \exists k \in \mathcal{N} \setminus \{n\} : \mathcal{Q}_n(\mathbf{x}_n(t)) \cap \mathcal{Q}_k(\mathbf{x}_k(t)) \neq \emptyset \right\}.$$

The conflicting drivable area $\mathcal{D}_n^C(\mathcal{X}_{n,0}, t)$ of the n -th vehicle is then the projection of its conflicting reachable set $\mathcal{R}_n^C(\mathcal{X}_{n,0}, t)$: $\mathcal{D}_n^C(\mathcal{X}_{n,0}, t) := \text{proj}(\mathcal{R}_n^C(\mathcal{X}_{n,0}, t))$.

Definition 4 (Conflict-Free Drivable Area): The conflict-free drivable area of the n -th vehicle is

$$\mathcal{D}_n^F(\mathcal{X}_{n,0}, t) := \mathcal{D}_n(\mathcal{X}_{n,0}, t) \setminus \mathcal{D}_n^C(\mathcal{X}_{n,0}, t).$$

The goal of our approach is the reallocation of $\mathcal{D}_n^C(\mathcal{X}_{n,0}, t)$ so that each cooperative traffic participant receives an individual area for motion planning. Henceforth, we will refer to the redistributed drivable area of the n -th vehicle as $\mathcal{D}_n^R(\mathcal{X}_{n,0}, t)$.

Definition 5 (Redistributed Drivable Area): We define the redistributed drivable area as

$$\mathcal{D}_n^R(\mathcal{X}_{n,0}, t) := \mathcal{D}_n^F(\mathcal{X}_{n,0}, t) \cup \mathcal{E}_n(\mathcal{X}_{n,0}, t),$$

where $\mathcal{E}_n(\mathcal{X}_{n,0}, t) \subseteq \mathcal{D}_n^C(\mathcal{X}_{n,0}, t)$. It holds that the interior of the redistributed drivable areas of all cooperative vehicles is pairwise disjoint.

III. BASIC IDEA

The starting point of our approach is the computation of the drivable areas $\mathcal{D}_n(\mathcal{X}_{n,0}, t)$ of each cooperative vehicle (see Fig. 2a). Since a conflict requires at least two different vehicles, the set of all conflicting subsets of vehicles is $\mathcal{P}_{\geq 2}(\mathcal{N})$, where $\mathcal{P}_{\geq 2}(\mathcal{N})$ denotes all subsets of the power set $\mathcal{P}(\mathcal{N})$ with cardinality greater than two. We demand that the vehicles staying in a conflict form a coalition ψ_r to solve it.

Definition 6 (Coalition): Let us introduce the relation $g : \{\mathcal{W}_1, \mathcal{W}_2, \dots, \mathcal{W}_e\} \rightarrow (\mathcal{W}_1, \mathcal{W}_2, \dots, \mathcal{W}_e)$ to convert the powerset $\mathcal{P}_{\geq 2}(\mathcal{N}) := \{\mathcal{W}_1, \mathcal{W}_2, \dots, \mathcal{W}_e\}$ into a tuple. The tuple $\Psi := g(\mathcal{P}_{\geq 2}(\mathcal{N}))$ is the ordered list of all unique subsets of vehicles which may have conflicting drivable areas. Henceforth, we refer to the r -th element $\psi_r \in \Psi$ as a coalition.

For instance, if we have $\mathcal{N} = \{1, 2, 3\}$, the tuple Ψ is:

$$\begin{aligned} \Psi &= g(\mathcal{P}_{\geq 2}(\mathcal{N})) \\ &= (\psi_1, \psi_2, \psi_3, \psi_4) \\ &= (\{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}). \end{aligned} \quad (1)$$

We assign a negotiable drivable area $\mathcal{D}^N(\psi_r, t)$ to each coalition ψ_r . To this end, we simplify the determination of $\mathcal{D}^N(\psi_r, t)$ and neglect the shape of the vehicles such that $\mathcal{Q}_n(\mathbf{x}_n(t)) = \mathbf{x}_n(t)$. This reduces the problem to the detection of overlapping regions (see Fig. 2b).

Definition 7 (Negotiable Drivable Area): We define the negotiable drivable area for each coalition $\psi_r \in \Psi$ as

$$\mathcal{D}^N(\psi_r, t) := \bigcap_{n \in \psi_r} \mathcal{D}_n^C(\mathcal{X}_{n,0}, t).$$

The members of the coalition ψ_r can distribute the negotiable drivable areas $\mathcal{D}^N(\psi_r, t)$ among each other. The redistribution of the sets $\mathcal{D}^N(\psi_r, t)$ can thereby be subject to a specific redistribution strategy, which e.g. minimizes a cost function J (see Fig. 2c).

The search for a feasible trajectory for each cooperative vehicle can be limited to its redistributed drivable area $\mathcal{D}_n^R(\mathcal{X}_{n,0}, t)$. This can reduce the computational complexity of multi-vehicle motion planning, since the n -th vehicle may only cause a collision with obstacles $\mathcal{O}(t)$ and other cooperative vehicles $\mathcal{N} \setminus \{n\}$ close to the border of $\mathcal{D}_n^R(\mathcal{X}_{n,0}, t)$. Especially, when one considers that a huge variety of combined trajectories of the collaborative vehicles can be excluded, since these combinations would lead to a collision. Our method reduces the set of trajectories which can be excluded, before motion planning. This can speed up the search for cooperative maneuvers.

Moreover, cooperative groups can be identified through $\mathcal{D}^N(\psi_r, \tau)$: if there exists a $\tau \in [0, t]$ such that $\mathcal{D}^N(\psi_r, \tau) \neq \emptyset$ holds, the vehicles belonging to the coalition ψ_r should plan their motion jointly.

IV. METHODOLOGY AND ALGORITHM

We apply an iterative approach which redistributes the drivable areas of the cooperative vehicles at discrete points in time t_i . During each iteration i , three steps have to be executed:

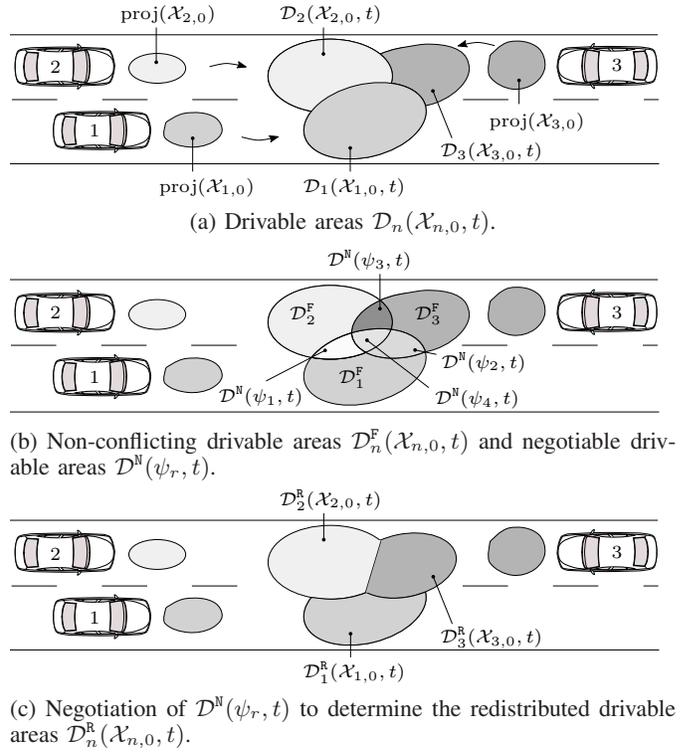


Fig. 2: Overview of the negotiation of drivable areas.

- 1) the computation of drivable areas $\mathcal{D}_n(\mathcal{X}_{n,i-1}, t_i)$,
- 2) the determination of negotiable areas $\mathcal{D}^N(\psi_r, t_i)$,
- 3) the redistribution of all negotiable areas $\mathcal{D}^N(\psi_r, t_i)$ to obtain $\mathcal{D}_n^R(\mathcal{X}_{n,i-1}, t_i)$. The new set of states $\mathcal{X}_{n,i}$ is computed based on $\mathcal{D}_n^R(\mathcal{X}_{n,i-1}, t_i)$.

There are three main challenges to be addressed: the efficient computation of the reachable set $\mathcal{R}_n(\mathcal{X}_{n,i-1}, t_i)$, the efficient assignment of the conflicting drivable areas $\mathcal{D}_n^C(\mathcal{X}_{n,i-1}, t_i)$ to a coalition ψ_r , and the choice of the redistribution strategy.

The reachable set $\mathcal{R}_n(\mathcal{X}_{n,i-1}, t_i)$, more specifically $\mathcal{D}_n(\mathcal{X}_{n,i-1}, t_i)$, cannot be computed efficiently for general system models $f_n(\mathbf{x}_n(t), \mathbf{u}_n(t))$ in the presence of arbitrary obstacles $\mathcal{O}(t)$. The overapproximation of the reachable set $\mathcal{R}_n(\mathcal{X}_{n,i-1}, t_i)$ constitutes a compromise between computational efficiency and accuracy; we use the approach presented in [19] (see Sec. IV-B).

Since we neglect the shape of the cooperative vehicles and model them as moving point masses, we are able to apply fast algorithms from the field of computational geometry to assign the conflicting drivable areas $\mathcal{D}_n^C(\mathcal{X}_{n,i-1}, t_i)$ to the coalitions ψ_r (see Sec. IV-C).

The choice of redistribution strategy highly influences the overall performance of the approach. In this paper, we select a strategy which should ensure that (see Sec. IV-D):

- the negotiable drivable areas $\mathcal{D}^N(\psi_r, t_i)$ are fairly allocated, such that all vehicles have equally sized drivable areas
- the redistributed drivable areas $\mathcal{D}_n^R(\mathcal{X}_{n,i-1}, t_i)$ are con-

nected.

The calculation of the drivable areas is based on models that are subsequently introduced and can be performed in a centralized or decentralized fashion. However, we assume that all computations are executed in a common coordinate system.

A. Vehicle and Obstacle Models

We model the dynamics of the cooperative vehicles as two double integrators with bounded velocity v and acceleration a . Let us introduce \square_x and \square_y to denote the value of the corresponding variable in x - and y -direction, respectively. After further introducing the notation $\underline{\square}$ and $\overline{\square}$ to specify the minimum and the maximum possible value of a variable, the dynamics is

$$\underbrace{\frac{d}{dt} \begin{bmatrix} s_{n,x} \\ v_{n,x} \\ s_{n,y} \\ v_{n,y} \end{bmatrix}}_{\mathbf{x}_n(t)} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} s_{n,x} \\ v_{n,x} \\ s_{n,y} \\ v_{n,y} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} a_{n,x} \\ a_{n,y} \end{bmatrix}}_{\mathbf{u}_n(t)}, \quad (2a)$$

$$\underline{v}_{n,x} \leq v_{n,x} \leq \overline{v}_{n,x}, \quad \underline{v}_{n,y} \leq v_{n,y} \leq \overline{v}_{n,y}, \quad (2b)$$

$$|a_{n,x}| \leq \overline{a}_{n,x}, \quad |a_{n,y}| \leq \overline{a}_{n,y}. \quad (2c)$$

The system dynamics of the obstacles is not restricted to any specific model. Moreover, we do not stipulate a particular method for the prediction of the occupancies of other traffic participants. Thus, any prediction supporting collision checks can be used, e.g. [20].

B. Computation of Drivable Areas

We compute the reachable set $\mathcal{R}_n(\mathcal{X}_{n,i-1}, t_i)$ of each cooperative vehicle at time t_i based on the set of states $\mathcal{X}_{n,i-1}$ of the previous time step t_{i-1} as presented in [19] and recall the main results of [19] in this subsection.

Pontryagin's principle is applied to calculate the boundaries of the reachable set of the n -th vehicle, ignoring the forbidden region $\mathcal{F}_n(t_i)$. Since the states of system model (2) are decoupled in longitudinal and lateral direction, their evolution over time can be computed separately. Two optimization problems are formulated in the x - and y -direction: given a specific terminal position, the objective is to maximize/minimize the speed at this position at time t . The Pontryagin principle yields a bang-bang input candidate solution with a single switching time. The velocity constraints (2b) are only considered at terminal time t . This results in an overapproximation of the reachable set, since the velocity constraints may be violated during $]t_{i-1}, t_i[$.

The lateral and longitudinal dynamics have to be considered jointly for the computation of $\mathcal{R}_n(\mathcal{X}_{n,i-1}, t_i)$, since it must be determined which states $\mathbf{x}_n(t_i)$ are within the set of forbidden states $\mathcal{F}_n(t_i)$. Söntges et. al [19] use the union of base sets $\mathcal{B}_{n,i}^{(q)}$, which are the Cartesian product of two convex polytopes in the (s_x, v_x) - and (s_y, v_y) -plane, to overapproximate $\mathcal{R}_n(\mathcal{X}_{n,i-1}, t_i)$ in a computationally efficient way:

$$\mathcal{R}_n(\mathcal{X}_{n,i-1}, t_i) \subseteq \bigcup_q \mathcal{B}_{n,i}^{(q)}.$$

The projection of the base sets $\mathcal{B}_{n,i}^{(q)}$ yields axis-aligned rectangles $\mathcal{A}_{n,i}^{(q)} := \text{proj}(\mathcal{B}_{n,i}^{(q)})$ in the position domain. It holds that the interior of the axis-aligned rectangles $\mathcal{A}_{n,i}^{(q)}$ is pairwise disjoint [19] and the union of rectangles $\mathcal{A}_{n,i}^{(q)}$ is an overapproximation of the drivable area $\mathcal{D}_n(\mathcal{X}_{n,i-1}, t_i)$:

$$\mathcal{D}_n(\mathcal{X}_{n,i-1}, t_i) \subseteq \bigcup_q \mathcal{A}_{n,i}^{(q)}.$$

C. Assignment of Drivable Areas to Coalitions

Next, the negotiable drivable areas are determined. We exploit the fact that the drivable areas $\mathcal{A}_{n,i} := \{\mathcal{A}_{n,i}^{(1)}, \mathcal{A}_{n,i}^{(2)}, \dots\}$ are represented by axis-aligned rectangles (see Fig. 3a) and use a sweep line algorithm [21], [22], known from the field of computational geometry, to conduct the following tasks:

- 1) the detection of overlapping axis-aligned rectangles (see Fig. 3b),
- 2) the division of the drivable areas $\bigcup_{n \in \mathcal{N}} \bigcup_q \mathcal{A}_{n,i}^{(q)}$ into the conflict-free and the negotiable drivable areas $\mathcal{A}_i^F := \{\mathcal{A}_{1,i}^F, \dots, \mathcal{A}_{N,i}^F\}$ and $\mathcal{A}_i^N := \{\mathcal{A}_{\psi_1,i}^N, \dots, \mathcal{A}_{\psi_l,i}^N\}$ of a single vehicle $n \in \mathcal{N}$ and a coalition $\psi_r \in \Psi$, respectively. All subsets $\mathcal{A}_{n,i}^F$ and $\mathcal{A}_{\psi_r,i}^N$ are a collection of novel axis-aligned rectangles $\mathcal{A}_{n,i}^{F(l)} \in \mathcal{A}_{n,i}^F$ and $\mathcal{A}_{\psi_r,i}^{N(p)} \in \mathcal{A}_{\psi_r,i}^N$, whose interior is disjoint (see Fig. 3c).

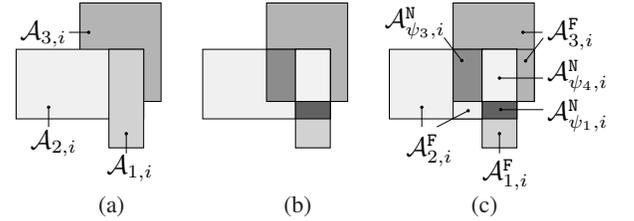


Fig. 3: Assignment of drivable areas at time step i : $\mathcal{A}_{n,i}^F$ and $\mathcal{A}_{\psi_r,i}^N$ represent the conflict-free drivable area of the n -th vehicle and the negotiable area of the coalition $\psi_r \in \Psi$ (1), respectively.

D. Negotiation of Drivable Areas

Finally, the negotiable drivable areas \mathcal{A}_i^N are redistributed. Our applied method is related to the *Nearest Centroid Classifier* [23]: given a set of distinct clusters with associated class labels, a new observation is classified according to the class label of the nearest cluster centroid.

Here, each cooperative vehicle is considered a single class; therefore, the set of class labels is \mathcal{N} . Furthermore, each connected component of the conflict-free drivable area $\mathcal{A}_{n,i}^F$ represents a cluster of the n -th vehicle. Thus, a single vehicle may have several clusters identified through the cluster centroid $\mathbf{c}_{n,i}^{(k)} := [x_{n,i}^{(k)}, y_{n,i}^{(k)}]^T$ with longitudinal and lateral position coordinate $x_{n,i}^{(k)}$ and $y_{n,i}^{(k)}$, respectively (see Fig. 4).

The connected components of the drivable area of a single vehicle are determined through a sweep line algorithm, which detects pairs of rectangles $\mathcal{A}_{n,i}^{F(l)}, \mathcal{A}_{n,i}^{F(o)} \in \mathcal{A}_{n,i}^F$, for $l \neq o$,

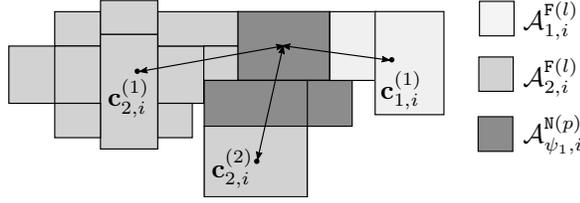


Fig. 4: The negotiable drivable areas are assigned to the vehicle with the nearest cluster centroid $\mathbf{c}_{n,i}^{(k)}$.

whose boundaries intersect [22], [24]. The cluster centroids are computed by geometric decomposition:

$$\begin{aligned} x_{n,i}^{(k)} &= \frac{\sum_l \text{centerX}(\mathcal{A}_{n,i}^{F(l)}) \text{area}(\mathcal{A}_{n,i}^{F(l)})}{\sum_l \text{area}(\mathcal{A}_{n,i}^{F(l)})}, \\ y_{n,i}^{(k)} &= \frac{\sum_l \text{centerY}(\mathcal{A}_{n,i}^{F(l)}) \text{area}(\mathcal{A}_{n,i}^{F(l)})}{\sum_l \text{area}(\mathcal{A}_{n,i}^{F(l)})}, \end{aligned} \quad (3)$$

where $\mathcal{A}_{n,i}^{F(l)}$ belongs to the k -th cluster of the n -th vehicle. The operators $\text{centerX}(\square)$, $\text{centerY}(\square)$, and $\text{area}(\square)$ return the x - and y -coordinates of the center and the area of an axis-aligned rectangle \square , respectively.

After agreeing that the set $\mathcal{C}_{n,i} := \{\mathbf{c}_{n,i}^{(1)}, \mathbf{c}_{n,i}^{(2)}, \dots\}$ contains all cluster centers $\mathbf{c}_{n,i}^{(k)}$ of the n -th vehicle at time step i , we denote the collection of all subsets $\mathcal{C}_{n,i}$ as $\mathcal{C}_i := \{\mathcal{C}_{1,i}, \dots, \mathcal{C}_{N,i}\}$. Subsequently, each negotiable rectangle $\mathcal{A}_{\psi_r,i}^{N(p)} \in \mathcal{A}_{\psi_r,i}^{N(p)}$ is assigned to the vehicle $n \in \psi_r$ with the nearest centroid $\mathbf{c}_{n,i}^{(k)}$ (see Fig. 4):

$$\arg \min_{n \in \psi_r} \left(\min_{\mathbf{c}_{n,i}^{(k)} \in \mathcal{C}_{n,i}} d(\mathcal{A}_{\psi_r,i}^{N(p)}, \mathbf{c}_{n,i}^{(k)}) \right), \quad (4)$$

where $d(\mathcal{A}_{\psi_r,i}^{N(p)}, \mathbf{c}_{n,i}^{(k)})$ denotes the Euclidean distance between the center of a rectangle $\mathcal{A}_{\psi_r,i}^{N(p)}$ and a cluster center $\mathbf{c}_{n,i}^{(k)}$.

E. Algorithm

Alg. 1 shows the overall approach. First, the overapproximated drivable area $\mathcal{A}_{n,i} := \{\mathcal{A}_{n,i}^{(1)}, \mathcal{A}_{n,i}^{(2)}, \dots\}$ of each cooperative vehicle is computed for each time step $i = 1, \dots, T$ by propagating the base sets $\mathcal{B}_{n,i-1} := \{\mathcal{B}_{n,i-1}^{(1)}, \mathcal{B}_{n,i-1}^{(2)}, \dots\}$ of the previous time step $i - 1$ under consideration of $\mathcal{F}_n(t_i)$ using the approach of [19] (Alg. 1, line 3).

Then, the coalitions $\psi_r \in \Psi$ and their corresponding negotiable areas \mathcal{A}_i^N , as well as the conflict-free areas \mathcal{A}_i^F , are determined in the function `COALITIONS` as explained in Sec. IV-C (Alg. 1, line 4).

The final step of our approach is the negotiation of the areas \mathcal{A}_i^N in the function `NEGOTIATE` (Alg. 1, line 8). The redistributed drivable areas $\mathcal{A}_i^R := \{\mathcal{A}_{1,i}^R, \dots, \mathcal{A}_{N,i}^R\}$ are initialized with the conflict-free drivable areas (see Alg. 1, line 9). Afterwards, we obtain the connected components as elaborated in Sec. IV-D and compute their corresponding cluster centroids with (3) (Alg. 1, line 10). Then, each coalition ψ_r negotiates

Algorithm 1

Input: Initial sets $\{\mathcal{B}_{1,0}, \dots, \mathcal{B}_{N,0}\}$, collision detection for axis-aligned rectangles with $\{\mathcal{F}_1(t), \dots, \mathcal{F}_N(t)\}$.

Output: $\{\mathcal{B}_{1,i}, \dots, \mathcal{B}_{N,i}\}$ for $i = 1, \dots, T$ time steps.

```

1: function DRIVINGAREAS( $\{\mathcal{B}_{1,0}, \dots, \mathcal{B}_{N,0}\}$ )
2:   for  $i = 1$  to  $T$  do
3:      $\{\mathcal{A}_{1,i}, \dots, \mathcal{A}_{N,i}\} \leftarrow \text{NEXTDRIVABLEAREAS}(\{
\mathcal{B}_{1,i-1}, \dots, \mathcal{B}_{N,i-1}\}
\{\mathcal{F}_1(t_i), \dots, \mathcal{F}_N(t_i)\})$ 
4:      $\mathcal{A}_i^F, \mathcal{A}_i^N \leftarrow \text{COALITIONS}(\{\mathcal{A}_{1,i}, \dots, \mathcal{A}_{N,i}\})$ 
5:      $\{\mathcal{B}_{1,i}, \dots, \mathcal{B}_{N,i}\} \leftarrow \text{NEGOTIATE}(\mathcal{A}_i^F, \mathcal{A}_i^N)$ 
6:   end for
7: end function

8: function NEGOTIATE( $\mathcal{A}_i^F, \mathcal{A}_i^N$ )
9:    $\mathcal{A}_i^R$ .initialize( $\mathcal{A}_i^F$ )
10:   $\mathcal{C}_i \leftarrow \text{COMPUTECLUSTERCENTROIDS}(\mathcal{A}_i^F)$ 
11:  for  $\mathcal{A}_{\psi_r,i}^N \in \mathcal{A}_i^N$  do
12:    if  $\mathcal{A}_{\psi_r,i}^N := \emptyset$  then
13:      continue
14:    end if
15:    for  $\mathcal{A}_{\psi_r,i}^{N(p)} \in \mathcal{A}_{\psi_r,i}^N$  do
16:       $n \leftarrow \text{ASSIGNMENT}(\mathcal{A}_{\psi_r,i}^{N(p)}, \mathcal{C}_i)$ 
17:       $\mathcal{A}_{n,i}^R := \mathcal{A}_{n,i}^R \cup \mathcal{A}_{\psi_r,i}^{N(p)}$ 
18:    end for
19:  end for
20:   $\{\mathcal{B}_{1,i}, \dots, \mathcal{B}_{N,i}\} \leftarrow \text{NEXTBASESETS}(\mathcal{A}_i^R)$ 
21:  return  $\{\mathcal{B}_{1,i}, \dots, \mathcal{B}_{N,i}\}$ 
22: end function

```

its associated area $\mathcal{A}_{\psi_r,i}^N$, provided that the set $\mathcal{A}_{\psi_r,i}^N$ is non-empty (Alg. 1, lines 12-14). Each rectangle $\mathcal{A}_{\psi_r,i}^{N(p)} \in \mathcal{A}_{\psi_r,i}^N$ is thereby assigned to the n -th vehicle through (4) in the function `ASSIGNMENT` (Alg. 1, line 16), where the Euclidean distance between the center of $\mathcal{A}_{\psi_r,i}^{N(p)}$ and all cluster centroids $\mathbf{c}_{n,i}^{(k)}$ of the vehicles $n \in \psi_r$ is computed. The n -th vehicle with the nearest cluster centroid receives the area, and its redistributed drivable area $\mathcal{A}_{n,i}^R$ is updated accordingly (Alg. 1, line 17). After the negotiation of the areas \mathcal{A}_i^N , the new base sets $\mathcal{B}_{n,i}$ are computed by adding the velocity information to the redistributed drivable areas $\mathcal{A}_{n,i}^R$, as presented in [19] (Alg. 1, line 20).

V. NUMERICAL EXAMPLES

We show the application of our algorithm on two highway scenarios: in the first example, we only consider collaborative vehicles, whereas in the second example, we take mixed traffic based on recorded traffic data into account.

A. Scenario I: Traffic with Self-Driving Vehicles

In our first example, we consider four collaborative vehicles as depicted in Fig. 5, which have to pass a narrow passage caused by two static obstacles. The parameters for the drivable

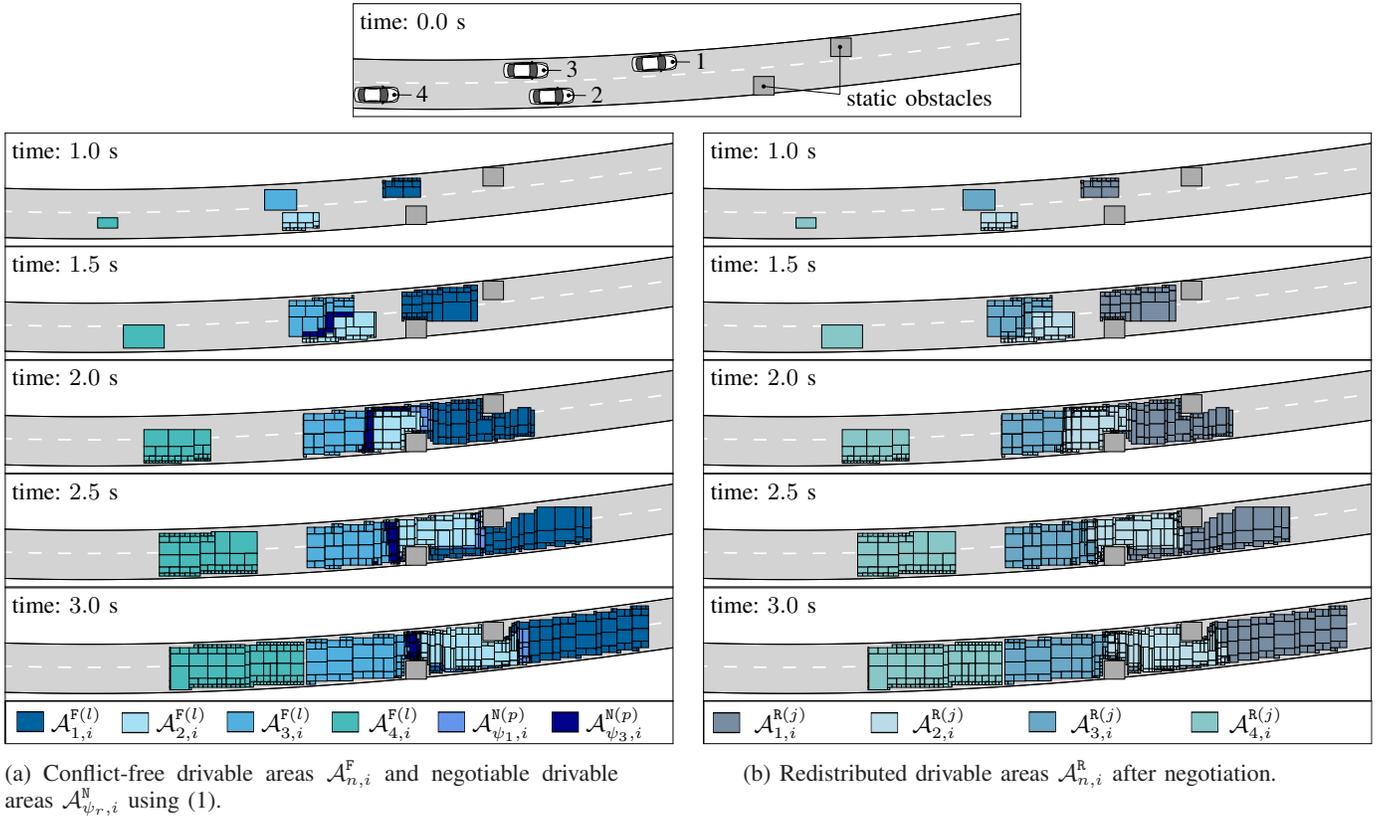


Fig. 5: Application of our proposed algorithm on a highway scenario with four collaborative vehicles.

area computation of vehicles 1 to 3 are selected equally, whereas vehicle 4 has less acceleration potential and the extreme values of the velocity are chosen to be different (see Tab. I).

Fig. 5a shows the conflict-free drivable areas $\mathcal{A}_{n,i}^F$ and the negotiable drivable areas $\mathcal{A}_{\psi_r,i}^N$ at different time steps i . Along the same lines, Fig. 5b depicts the redistributed drivable areas $\mathcal{A}_{n,i}^R$ of each collaborative vehicle at the same time steps i . It can be determined that the movement of vehicle 4 is temporally conflict-free, since its drivable area does not intersect with any drivable area of the other vehicles. Thus, cooperation is unnecessary, and the trajectory planning for vehicle 4 can be conducted exclusively.

Furthermore, it can be identified that the drivable areas of vehicles 2 and 3 are in conflict first (see Fig. 5a at time instance 1.5 s). The conflict is resolved by giving vehicle 2 precedence over vehicle 3 to pass the obstacle in its current lane, which is automatically initiated by our approach. At a later stage, the drivable areas of vehicle 1 and 2 overlap; however, there is no change in the driving strategy, meaning that vehicle 1 stays in front of vehicle 2.

B. Scenario II: Mixed Traffic

We further demonstrate the applicability of our algorithm on the mixed-traffic scenario C-NGSIM_US101_1 based on the NGSIM US 101 Highway Dataset¹ (7:50 a.m. to 8:05 a.m.)

TABLE I

Scenario I: Parameters for drivable area computation.

Parameter	Value
time discretization	0.1 s
time steps T	30
maximum speed $\bar{v}_{1/2/3,x}$	22.0 m/s
minimum speed $\underline{v}_{1/2/3,x}$	0.0 m/s
absolute maximum speed ($\bar{v}_{1/2/3,y} = -\underline{v}_{1/2/3,y}$)	15.0 m/s
absolute maximum acceleration $\bar{a}_{1/2/3,x}$	8.0 m/s ²
absolute maximum acceleration $\bar{a}_{1/2/3,y}$	4.0 m/s ²
maximum speed $\bar{v}_{4,x}$	18.0 m/s
minimum speed $\underline{v}_{4,x}$	0.0 m/s
absolute maximum speed ($\bar{v}_{4,y} = -\underline{v}_{4,y}$)	7.0 m/s
absolute maximum acceleration $\bar{a}_{4,x}$	4.0 m/s ²
absolute maximum acceleration $\bar{a}_{4,y}$	2.0 m/s ²

from the CommonRoad² benchmark collection [25]. As shown in Fig. 7, there are four collaborative vehicles for which we compute the drivable areas based on the parameters given in Tab. II. The drivable areas of the vehicles are thereby restricted to five lanes, since we do not consider the highway on-ramp. We use the recorded trajectory data for the prediction of the obstacle movement and enlarge their rectangular shape to simulate uncertainty. However, any prediction which supports collision checks with axis-aligned rectangles can be applied.

¹<http://www.fhwa.dot.gov/publications/research/operations/07030/>

²<http://commonroad.in.tum.de>

TABLE II

Scenario II: Parameters for the drivable area computation.

Parameter	Value
time discretization	0.1 s
time steps T	30
maximum speed $\bar{v}_{n,x}$	36.0 m/s
minimum speed $\underline{v}_{n,x}$	0.0 m/s
absolute maximum speed ($\bar{v}_{n,y} = -\underline{v}_{n,y}$)	7.0 m/s
absolute maximum acceleration $\bar{a}_{n,x}$	5.5 m/s ²
absolute maximum acceleration $\bar{a}_{n,y}$	2.5 m/s ²

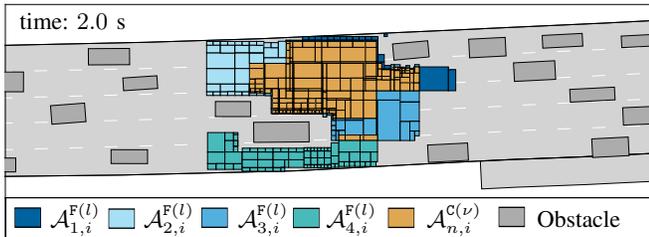


Fig. 6: Huge parts of the drivable areas of the collaborative vehicles overlap if the conflicting areas $\mathcal{A}_{n,i}^{C(\nu)}$ are not negotiated.

We first analyze if there is the need for cooperation and compute the drivable areas of each collaborative vehicle using the approach of [19]. As shown in Fig. 6, the overapproximated drivable areas of vehicles 1, 2, and 3 overlap almost entirely at time instance 2.0 s (see $\mathcal{A}_{n,i}^{C(\nu)}$ in Fig. 6). This indicates that a vast majority of combined trajectories of the collaborative vehicles lead to a collision among each other.

In contrast, when applying our proposed method, the propagated drivable areas are negotiated at each time step i such that they do not overlap. Moreover, high-level driving strategies are determined before motion planning (see Fig. 7): Vehicle 1 merges in front of vehicle 3. Vehicle 4 follows vehicle 3, whereas vehicle 2 may stay behind vehicle 3 or may drive behind vehicle 4 (see Fig. 7 at time instance 3.0 s). As can be seen, our approach works for complex traffic scenarios and efficiently distributes the drivable areas.

VI. CONCLUSIONS

We present a new approach for cooperative conflict resolution based on the negotiation of drivable areas, wherein the computation of drivable areas is based on reachability analysis. Our method is suitable for reducing the search space for motion planning given the assumption that the admissible trajectories of a cooperative vehicle must be contained in its redistributed drivable area. An important aspect of our method is the independence of specific obstacle representations. In the future, we plan to investigate different negotiation strategies to extend our approach to arbitrary road networks. Furthermore, we want to integrate individual objectives during the negotiation phase, e.g. by using auction-based approaches [11], [26].

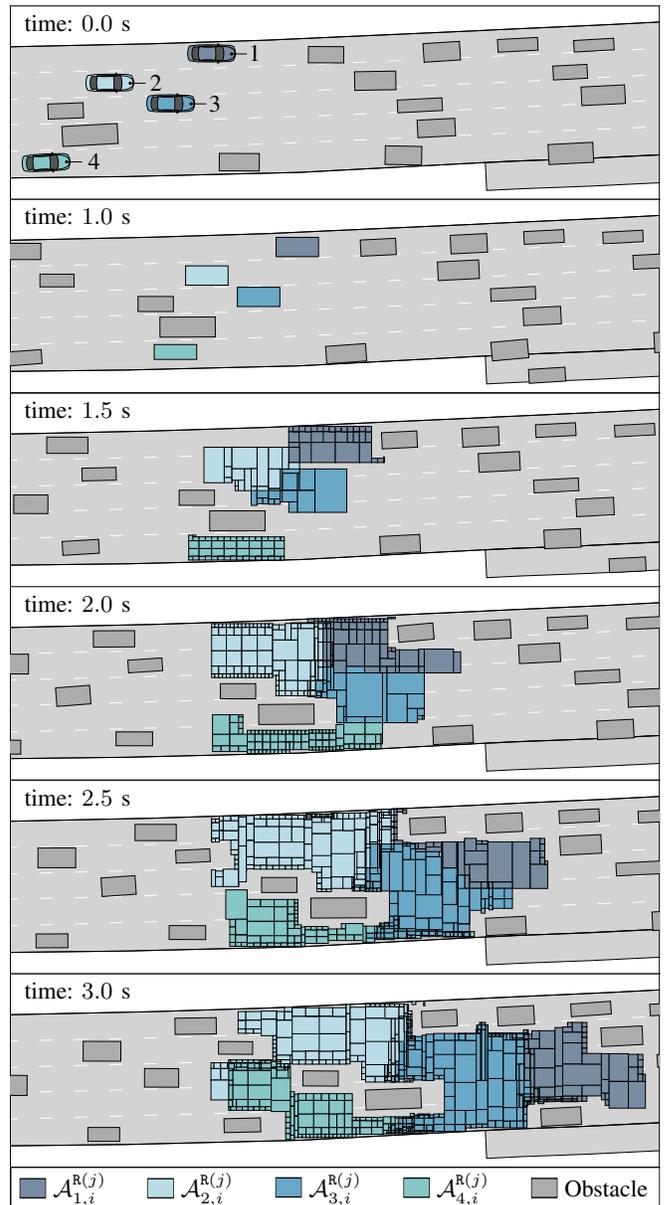


Fig. 7: Redistributed drivable areas $\mathcal{A}_{n,i}^R$ at different points in time t_i .

ACKNOWLEDGMENT

This work was supported by the Deutsche Forschungsgemeinschaft (German Research Foundation) within the Priority Program SPP 1835 Cooperative Interacting Automobiles (grant number: AL 1185/4-1).

REFERENCES

- [1] M. R. Hafner, D. Cunningham, L. Caminiti, and D. Del Vecchio, "Cooperative collision avoidance at intersections: Algorithms and experiments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1162–1175, 2013.
- [2] A. Colombo and D. Del Vecchio, "Least restrictive supervisors for intersection collision avoidance: A scheduling approach," *IEEE Transactions on Automatic Control*, vol. 60, no. 6, pp. 1515–1527, 2015.

- [3] D. Caveney and W. B. Dunbar, "Cooperative driving: Beyond V2V as an ADAS sensor," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2012, pp. 529–534.
- [4] M. Goli and A. Eskandarian, "Evaluation of lateral trajectories with different controllers for multi-vehicle merging in platoon," in *International Conference on Connected Vehicles and Expo*, 2014, pp. 673–678.
- [5] H. C.-H. Hsu and A. Liu, "Kinematic design for platoon-lane-change maneuvers," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 1, pp. 185–190, 2008.
- [6] C. Frese and J. Beyerer, "A comparison of motion planning algorithms for cooperative collision avoidance of multiple cognitive automobiles," in *IEEE Intelligent Vehicles Symposium*, 2011, pp. 1156–1162.
- [7] K. E. Bekris, K. I. Tsianos, and L. E. Kavraki, "Safe and distributed kinodynamic replanning for vehicular networks," *Mobile Networks and Applications*, vol. 14, no. 3, pp. 292–308, 2009.
- [8] J. Alonso, V. Milanés, J. Pérez, E. Onieva, C. González, and T. de Pedro, "Autonomous vehicle control systems for safe crossroads," *Transportation research part C: emerging technologies*, vol. 19, no. 6, pp. 1095–1110, 2011.
- [9] J. Grégoire, S. Bonnabel, and A. de La Fortelle, "Priority-based intersection management with kinodynamic constraints," in *European Control Conference*, 2014, pp. 2902–2907.
- [10] M. G. Plessen, D. Bernardini, H. Esen, and A. Bemporad, "Multi-automated vehicle coordination using decoupled prioritized path planning for multi-lane one- and bi-directional traffic flow control," in *IEEE 55th Conference on Decision and Control*, 2016, pp. 1582–1588.
- [11] H. Rewald and O. Stursberg, "Cooperation of autonomous vehicles using a hierarchy of auction-based and model-predictive control," in *IEEE Intelligent Vehicles Symposium*, 2016, pp. 1078–1084.
- [12] M. Vasirani and S. Ossowski, "A market-inspired approach for intersection management in urban road traffic networks," *Journal of Artificial Intelligence Research*, vol. 43, pp. 621–659, 2012.
- [13] H. Schepperle and K. Böhm, "Auction-based traffic management: Towards effective concurrent utilization of road intersections," in *10th IEEE Conference on E-Commerce Technology and the Fifth IEEE Conference on Enterprise Computing, E-Commerce and E-Services*, 2008, pp. 105–112.
- [14] D. Carlino, S. D. Boyles, and P. Stone, "Auction-based autonomous intersection management," in *16th International IEEE Conference on Intelligent Transportation Systems*, 2013, pp. 529–534.
- [15] K. Dresner and P. Stone, "Turning the corner: improved intersection control for autonomous vehicles," in *IEEE Intelligent Vehicles Symposium*, 2005, pp. 423–428.
- [16] O. Mehani and A. de La Fortelle, "Trajectory planning in a crossroads for a fleet of driverless vehicles," in *Proc. of the 11th International Conference on Computer Aided Systems Theory*. Springer Berlin Heidelberg, 2007, pp. 1159–1166.
- [17] A. de La Fortelle, "Analysis of reservation algorithms for cooperative planning at intersections," in *13th International IEEE Conference on Intelligent Transportation Systems*, 2010, pp. 445–449.
- [18] D. Marinescu, J. Čurn, M. Bouroche, and V. Cahill, "On-ramp traffic merging using cooperative intelligent vehicles: A slot-based approach," in *15th International IEEE Conference on Intelligent Transportation Systems*, 2012, pp. 900–906.
- [19] S. Söntges and M. Althoff, "Computing the drivable area of autonomous road vehicles in dynamic road scenes," *IEEE Transactions on Intelligent Transportation Systems*, [to appear].
- [20] M. Althoff and S. Magdici, "Set-based prediction of traffic participants on arbitrary road networks," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 2, pp. 187–202, 2016.
- [21] J. L. Bentley and T. A. Ottmann, "Algorithms for reporting and counting geometric intersections," *IEEE Transactions on Computers*, vol. C-28, no. 9, pp. 643–647, 1979.
- [22] R. G. Karlsson and M. H. Overmars, "Scanline algorithms on a grid," *BIT Numerical Mathematics*, vol. 28, no. 2, pp. 227–241, 1988.
- [23] E.-H. S. Han and G. Karypis, *Centroid-Based Document Classification: Analysis and Experimental Results*. Springer Berlin Heidelberg, 2000, pp. 424–431.
- [24] H. Edelsbrunner, J. Van Leeuwen, T. Ottmann, and D. Wood, "Computing the connected components of simple rectilinear geometrical objects in d -space," *RAIRO, Informatique théorique*, vol. 18, no. 2, pp. 171–183, 1984.
- [25] M. Althoff, M. Koschi, and S. Manziinger, "CommonRoad: Composable benchmarks for motion planning on roads," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2017, pp. 719–726.
- [26] M. B. Dias, R. Zlot, N. Kalra, and A. Stentz, "Market-based multirobot coordination: A survey and analysis," *Proc. of the IEEE*, vol. 94, no. 7, pp. 1257–1270, 2006.