Tactical Decision Making for Cooperative Vehicles Using Reachable Sets

Stefanie Manzinger and Matthias Althoff

Abstract— **Tactical maneuver planning of multiple, communicating vehicles provides the opportunity to increase passenger safety and comfort. We propose a unifying method to orchestrate the motion of cooperative vehicles based on the negotiation of conflicting road areas, which are determined by reachable set computation. As a result, each vehicle receives an individual driving corridor for trajectory planning. The presented conflict resolution scheme has polynomial runtime complexity and is guaranteed to find the optimal allocation of road areas for each negotiation round. Our method is not tailored to specific traffic situations but is applicable to general traffic scenes with manually driven and automated vehicles. We demonstrate the universal usability of our approach in numerical experiments.**

I. Introduction

There are many traffic situations in which individual navigation goals of traffic participants lead to conflicts. Human drivers often resolve these issues through implicit communication relying on the reasonable behavior of others. However, communicating automated vehicles offer more sophisticated solutions for collaborative maneuver planning. These vehicles can form a cooperative group, which jointly agrees on a common driving strategy to achieve conflict avoidance while maximizing individual utilities. One of the major challenges towards multi-vehicle motion planning is the development of coordination schemes which are computationally efficient without compromising optimality. We introduce a method for tactical decision making, which unambiguously assigns road areas to cooperating vehicles. Subsequently, we review literature focusing on similar conflict resolution principles.

Dresner et al. [1] pioneered the work on reservation-based algorithms with a focus on intersection management: the intersection space is discretized into tiles, which approaching vehicles can request via an intersection manager. This manager simulates trajectories of vehicles to determine the necessary tiles for passing the intersection and ensures that no tile is occupied by more than one vehicle. The original work has been extended in successive publications [2]–[4]. Sharon et al. [5] improve on the work of Dresner et al. [4] to be more efficient, particularly if the majority of vehicles present are driven by humans. In [6]–[8], the first-come, first-served policy for reservation assignment in [1] is replaced by auction-based methods. In [9], [10], conflict points instead of tiles are used as a resource for intersection management; Levin et al. [11] combine conflict points at intersections into conflict regions. In [12], a legacy algorithm based on reservations is proposed, which can handle a low percentage of non-communicating vehicles or vehicles with malfunctioning communication systems. A comprehensive overview of further techniques for cooperative intersection management can be found in [13].

While the former methods are applied to intersections, Marinescu et al. [14], [15] present virtual slots for traffic shaping. A virtual slot is a moving space-time corridor with a predefined behavior, e.g., lane following with constant speed or lane changing; vehicles assigned to slots must adopt their behavior. Their method is evaluated on a highway merging scenario. Alternative solutions to cooperative on-ramp merging onto highways can be found in [16]. Zhang et al. [17] propose a reservation-based scheduling technique to coordinate communicating vehicles through an intersection, which is divided into a set of static critical sections. Their goal is to establish a service-orientated traffic management, where high-priority vehicles are able to pass through the intersection first. In [18], dynamic critical sections are introduced among static critical sections to consider behaviors like lane-changing and overtaking. Cooperative maneuvers are defined as a sequence of states modeled within an event-triggered state automaton [18].

We intend to solve temporarily bounded conflicts, where vehicles collaborate for a limited amount of time based on reservations: road areas requested by several communicating traffic participants are distributed such that each vehicle receives its own driving corridor for trajectory planning. In contrast to previous work, we determine reservation conflicts by computing the drivable areas of all collaborative vehicles using reachability analysis. While many works assume that vehicles are highly automated, we intentionally deal with mixed traffic, where human-driven and automated vehicles share the road. Since scenarios with only automated vehicles are a special case, our algorithm can treat these situations as well. Moreover, our unifying method is not restricted to specific maneuvers or traffic situations.

Our approach can be categorized as a hybrid framework [13], where we optimally use the capabilities of each vehicle through decentralized computation of reachable sets, while the negotiation is performed by a leading vehicle. The presented paper is based on our previous work in [19]. The novelty of this work includes:

- Designing a general framework to incorporate individual goals of cooperative vehicles, whereas in [19], the value assessment of different road areas is exclusively based on geometric reasoning.

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

All authors are with the Department of Computer Science, Technical University of Munich, 85748 Garching bei München, Germany {stefanie.manzinger, althoff}@in.tum.de

- Formulating a combinatorial optimization problem to optimally allocate conflicting road areas to the vehicles according to their preferences and reducing its computational complexity through hierarchical structuring of conflicting road areas. This makes it possible to resolve conflicts in polynomial runtime complexity in the number of road areas to be negotiated [20], but independently in the number of vehicles.
- Computing the reachable sets of cooperative traffic participants in a vehicle-specific, curvilinear coordinate system, whereas in [19], it is required that all computations are performed in a common coordinate system. This facilitates coping with different driving contexts as presented in Sec. VI.

Sec. II introduces the problem statement and Sec. III presents the necessary preliminaries. In Sec. IV-V, our applied methods and proposed algorithm are described. Sec. VI demonstrates our approach on numerical examples, followed by the conclusion in Sec.VII.

II. Problem Statement

Let $\mathcal{G} = \{g_0, g_1, \dots, g_i, \dots\}$ denote a grid with cells g_i of an arbitrary shape obtained through tessellation of the position domain in a Cartesian reference frame F_0 (see Fig. 1). The cells g_i are the individual assets of road areas which can be combined into unions of assets $\mathcal{C}_j \subseteq \mathcal{G}$, which we refer to as packages. We introduce the set $\mathcal{V} := \{V_1, V_2, \dots, V_N\}$ of cooperative vehicles acting as bidders, which can submit a bid for different sets $\mathcal{C}_j$. We restrict the set of permitted combinations $\mathcal{P}(t) \subseteq \mathtt{P}(\mathcal{G}^{\mathrm{c}}(t))$, where $\mathtt{P}()$ returns the powerset, to those $\mathcal{C}_j$ containing only conflicting cells $g_i \in \mathcal{G}^{\mathrm{c}}(t)$, $\mathcal{G}^{\mathrm{c}}(t) \subseteq \mathcal{G}$, requested from at least two vehicles V_n at time instance t (see Fig. 1). It is assumed that each vehicle V_n only bids its true value $b_n(t, \mathcal{C}_j)$ of a combination of assets $\mathcal{C}_j$. The maximum bid for a package $\mathcal{C}_j$ is $\overline{b}(t, \mathcal{C}_j)$, and any tie-breaking rule [20] is accepted to determine $\overline{b}(t, \mathcal{C}_j)$.

We aim to find a distribution of sets $\mathcal{C}_j \in \mathcal{P}(t)$ such that the revenue is maximized (1a) and no single asset is assigned more than once (1b) [20]:

$$\max_{\delta(t,\mathcal{C}_j)} \sum_{\mathcal{C}_j \in \mathcal{P}(t)} \delta(t, \mathcal{C}_j) \overline{b}(t, \mathcal{C}_j) \tag{1a}$$

such that

$$\forall g_i \in \mathcal{G}^{\mathrm{c}}(t): \sum_{\mathcal{C}_j:\ g_i \in \mathcal{C}_j} \delta(t, \mathcal{C}_j) \leq 1, \tag{1b}$$

$$\forall \mathcal{C}_j \in \mathcal{P}(t):\ \delta(t, \mathcal{C}_j) \in \{0, 1\}, \tag{1c}$$

where $\delta(t, \mathcal{C}_j)$ denotes the allocation of package $\mathcal{C}_j$ to the highest bidder V_n; $\delta(t, \mathcal{C}_j) = 1$ holds iff bidder V_n receives package $\mathcal{C}_j$ at time instance t.

The optimization problem (1) is known as the *winner determination problem*, which is NP-hard to solve [20], [21]. Moreover, allowing every possible combination of assets $g_i \in \mathcal{G}^{\mathrm{c}}(t)$ means that each bidder V_n has to evaluate $2^{|\mathcal{G}^{\mathrm{c}}(t)|} - 1$ packages. However, we are able to attain computational tractability by selecting a special structure of permitted combinations $\mathcal{P}(t)$: we require $\mathcal{P}(t)$ to form a tree structure; thus, an optimal allocation of packages can be found in $O(|\mathcal{G}^{\mathrm{c}}(t)|^2)$ time [20].

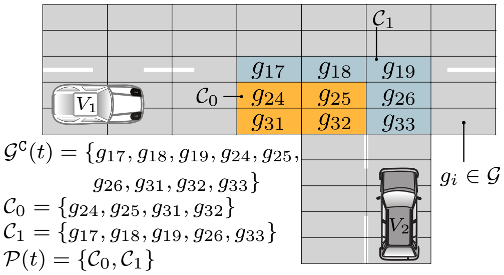

$\mathcal{C}_1$
g_{17}
g_{18}
g_{19}
V_1
$\mathcal{C}_0$
g_{24}
g_{25}
g_{26}
g_{31}
g_{32}
g_{33}
$\mathcal{G}^{\mathrm{c}}(t) = \{g_{17}, g_{18}, g_{19}, g_{24}, g_{25}, g_{26}, g_{31}, g_{32}, g_{33}\}$
$g_i \in \mathcal{G}$
$\mathcal{C}_0 = \{g_{24}, g_{25}, g_{31}, g_{32}\}$
$\mathcal{C}_1 = \{g_{17}, g_{18}, g_{19}, g_{26}, g_{33}\}$
V_2
$\mathcal{P}(t) = \{\mathcal{C}_0, \mathcal{C}_1\}$

Fig. 1. Visualization of the grid $\mathcal{G}$, the set of conflicting cells $\mathcal{G}^{\mathrm{c}}(t)$, the packages $\mathcal{C}_j$, and the set of permitted combinations $\mathcal{P}(t)$.

III. Preliminaries

We describe the system dynamics of the n-th cooperative vehicle in a local curvilinear coordinate system F_n by the differential equation

$$\dot{x}_n(t) = f_n(x_n(t), u_n(t)), \tag{2}$$

where $x_n \in \mathcal{X}_n \subseteq \mathbb{R}^p$ is the state, $u_n \in \mathcal{U}_n \subset \mathbb{R}^q$ is the input, and t is the time. We denote the solution of (2) for the input trajectory $u_n(\cdot)$ and initial state $x_n(0)$ with $\chi_n(t; x_n(0), u_n(\cdot))$. Please note that each vehicle may have its own coordinate system F_n.

The reachable set is defined as the set of all states which can be reached from an initial set of states $\mathcal{X}_{n,0}$ at a given time instance t. We extend this standard definition and restrict the reachable set $\mathcal{R}_n(\mathcal{X}_{n,0}, t)$ of vehicle V_n to all collision-free reachable states. We therefore introduce the set of forbidden states $\mathcal{F}_n(t) = \{x_n(t) | \mathcal{Q}_n(x_n(t)) \cap \mathcal{O}_n(t) \neq \emptyset\}$, where $\mathcal{Q}_n(x_n(t)) \subset \mathbb{R}^2$ and $\mathcal{O}_n(t) \subset \mathbb{R}^2$ denote the occupied space of the ego vehicle and the (time-varying) obstacles, respectively. Thus,

$$\begin{aligned}\mathcal{R}_n(\mathcal{X}_{n,0}, t) = \Big\{ \chi_n(t; x_n(0), u_n(\cdot)) \Big| x_n(0) \in \mathcal{X}_{n,0}, \\ \forall \tau \in [0, t]: u_n(\tau) \in \mathcal{U}_n, \chi_n(\tau; x_n(0), u_n(\cdot)) \notin \mathcal{F}_n(\tau) \Big\}.\end{aligned} \tag{3}$$

Moreover, we specify the relation $h_n: \mathtt{P}(\mathcal{X}_n) \to \mathtt{P}(\mathcal{G})$, which returns the cells $g_i \in \mathcal{G}$ occupied by vehicle V_n due to its set of states $\mathcal{X}_n$ considering its shape, to determine the set $\mathcal{G}^{\mathrm{c}}(t)$ of conflicting cells claimed by multiple vehicles:

$$\mathcal{G}^{\mathrm{c}}(t) = \bigcup_{\mathcal{I} \in \mathtt{P}_{\geq 2}(\mathcal{N})} \bigcap_{n \in \mathcal{I}} h_n(\mathcal{R}_n(\mathcal{X}_{n,0}, t)), \tag{4}$$

where $\mathtt{P}_{\geq 2}(\mathcal{N})$ denotes all subsets of the power set $\mathtt{P}(\mathcal{N})$ with cardinality greater than one and $\mathcal{N} := \{1, 2, \dots, N\}$. We thereby assume the forward and backward transformation from F_n to F_0 to be given, see e.g. [22]. We further introduce the negotiated reachable set of vehicle V_n at time t:

$$\mathcal{R}^{\mathtt{N}}_n(\mathcal{X}_{n,0},t)=\Big\{x_n(t)\in\mathcal{R}_n(\mathcal{X}_{n,0},t)\Big| \\ h_n(\{x_n(t)\})\cap\mathcal{G}^{\mathtt{L}}_n(t)=\emptyset\Big\}, \tag{5}$$

where $\mathcal{G}^{\mathtt{L}}_n(t)\subseteq\mathcal{G}^{\mathtt{C}}(t)$ denotes the set of unassigned grid cells g_i of vehicle V_n after the negotiation (1).

IV. Conflict Resolution

Conflict resolution is performed at discrete time steps k, which correspond to points in time $t_k=k\Delta t$, where $\Delta t\in\mathbb{R}^+$ is a constant time step. We identify the individual driving areas of vehicles V_n iteratively for each time step k by applying Alg. 1, which comprises the following steps:

1) computation of the reachable sets (3) (Alg. 1, line 3),
2) identification of conflicting cells $\mathcal{G}^{\mathtt{C}}(k)$ using (4) (Alg. 1, line 4),
3) negotiation of conflicting cells $\mathcal{G}^{\mathtt{C}}(k)$ to determine the optimal allocation $\mathcal{W}_{\mathtt{opt}}$ of $g_i\in\mathcal{G}^{\mathtt{C}}(k)$ (Alg. 1, line 5),
4) determination of the negotiated reachable sets (5) (Alg. 1, line 6).

Below, we elaborate steps 1) and 3) comprehensively and use the notation $[\square_n]_{n=1}^N=[\square_1,\dots,\square_n,\dots,\square_N]$ to denote a list of elements $\square_n$ of vehicles V_n.

Algorithm 1

```
1: function ConflictResolution($[\mathcal{B}^{\mathtt{N}}_n(0)]_{n=1}^N$, $\mathcal{G}$)
2:     for $k=1$ to $T$ do
3:         $[\mathcal{B}_n(k)]_{n=1}^N \leftarrow$ReachableSets($[\mathcal{B}^{\mathtt{N}}_n(k-1)]_{n=1}^N$)
4:         $\mathcal{G}^{\mathtt{C}}(k) \leftarrow$ConflictingCells($[\mathcal{B}_n(k)]_{n=1}^N$, $\mathcal{G}$)
5:         $\mathcal{W}_{\mathtt{opt}} \leftarrow$Negotiate($[\mathcal{B}_n(k)]_{n=1}^N$, $\mathcal{G}^{\mathtt{C}}(k)$)
6:         $[\mathcal{B}^{\mathtt{N}}_n(k)]_{n=1}^N \leftarrow$ NegotiatedReachableSets(
                                   $[\mathcal{B}_n(k)]_{n=1}^N, \mathcal{W}_{\mathtt{opt}}$)
7:     end for
8:     return $[\cup_k\mathcal{B}^{\mathtt{N}}_n(k)]_{n=1}^N$
9: end function
```

A. Reachable Set Computation

1) Vehicle Dynamics: We model the dynamics of vehicle V_n in the local coordinate system F_n as two double integrators in longitudinal ζ_n- and lateral η_n-direction with bounded speed v_n and acceleration u_n. After introducing the notation $\underline{\square}$ and $\overline{\square}$ to specify the minimum and the maximum possible value of a variable $\square$, the dynamics is

$$\ddot{s}_{n,\zeta_n}(t)=u_{n,\zeta_n}(t),\qquad \ddot{s}_{n,\eta_n}(t)=u_{n,\eta_n}(t), \tag{6a}$$

$$\underline{v}_{n,\zeta_n}\le v_{n,\zeta_n}(t)\le\overline{v}_{n,\zeta_n},\quad \underline{v}_{n,\eta_n}\le v_{n,\eta_n}(t)\le\overline{v}_{n,\eta_n}, \tag{6b}$$

$$|u_{n,\zeta_n}(t)|\le\overline{a}_{n,\zeta_n},\quad |u_{n,\eta_n}(t)|\le\overline{a}_{n,\eta_n}, \tag{6c}$$

where $s_{n,\zeta_n}(t)$ and $s_{n,\eta_n}(t)$ denote the position in longitudinal and lateral direction, respectively.

Model (6) is an approximation of the real vehicle dynamics, which deviates increasingly from a real vehicle the larger the curvature of the road; the modeled vehicle would be able to make a turn with an arbitrarily high velocity. However, we compensate for this by setting appropriate constraints (6b)-(6c). The use of a curvilinear coordinate frame facilitates the formulation of certain properties and maneuvers, which are highly relevant to our approach, e.g., lane-following, stopping at an intersection, and avoiding driving backwards.

2) Reachable Set: Reachable sets are computed according to [23]: we approximate the reachable set at time step k by the union of base sets $\mathcal{B}^{(i)}_n(k)$, which are composed of the Cartesian product of two convex polytopes in the $(s_{n,\zeta_n},v_{n,\zeta_n})$- and $(s_{n,\eta_n},v_{n,\eta_n})$-plane:

$$\mathcal{R}_n(\mathcal{B}^{\mathtt{N}}_n(k-1),t_k)\approx\bigcup_i\mathcal{B}^{(i)}_n(k)=:\mathcal{B}_n(k),$$

where $\mathcal{B}^{\mathtt{N}}_n(k-1)$ denotes the negotiated reachable sets of the previous time step $k-1$. The projection of base sets $\mathcal{B}^{(i)}_n(k)$ in the position domain—in this paper referred to as drivable area—yields axis-aligned rectangles.

The reachable sets are computed with reference to the center of gravity of the vehicle; however, we need to consider the shapes of vehicles for collision detection. Söntges et al. [23] use the inner circle of the vehicle shape for collision detection, whereas we approximate the shape of vehicles V_n with three rotationally invariant disks with radius r_n and assume that the heading of the vehicles is aligned with ζ_n in F_n. Please note that the reachable set computation is not overapproximative due to the modified collision checks.

B. Negotiation of Conflicting Cells

Our negotiation scheme for resolving conflicts between collaborative vehicles is inspired by the idea of auctions. An auction requires a set of buyers competing for limited resources; bids are used to express preferences over the auctioned resources. In this work, cooperative vehicles act as bidders, and the limited resource is the drivable space on the road. In contrast to general auction design, we do not introduce a pricing mechanism [21]. Furthermore, we apply a common utility function to determine the bids for each vehicle. The negotiation of conflicting road cells comprises the following steps (see Alg. 2):

1) determination of permitted packages $\mathcal{C}_j\in\mathcal{P}(k)$ on the basis of $\mathcal{G}^{\mathtt{C}}(k)$ and their structuring in a tree $\mathcal{T}$ (see Fig. 2b, Alg. 2, line 4),
2) evaluation of bids $\bar{b}(k,\mathcal{C}_j)$ for $\mathcal{C}_j\in\mathcal{P}(k)$ (Alg. 2, line 5),
3) computation of the optimal allocation $\mathcal{W}_{\mathtt{opt}}$ of permitted packages $\mathcal{C}_j\in\mathcal{P}(k)$ (Alg. 2, line 6).

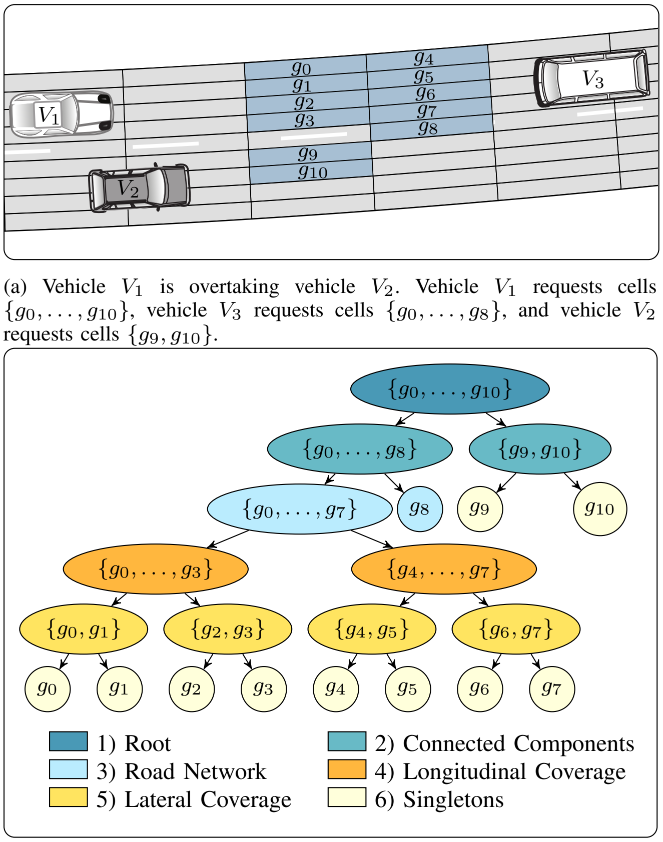
In the remainder of this section, all necessary steps are discussed in detail.

1) Tree Structure: The entire combination $\mathcal{G}^{\mathtt{C}}(k)$ including all negotiable assets is the root node of the tree (see Fig. 2b). At each level of the tree, we decompose the sets $\mathcal{C}_j$ into disjoint parts, of which each part represents a set $\mathcal{C}_l\in\mathcal{P}(k)$. Thus, for all $\mathcal{C}_j,\mathcal{C}_i\in\mathcal{P}(k)$, we have $\mathcal{C}_j\cap\mathcal{C}_i\in\{\emptyset,\mathcal{C}_j,\mathcal{C}_i\}$ [20] (see Fig. 2b).

As mentioned in Sec. II, using a tree structure to express the set of permitted combinations $\mathcal{P}(k)$ of cells simplifies

Algorithm 2

```
function NEGOTIATE($[\mathcal{B}_n(k)]_{n=1}^N$, $\mathcal{G}^c(k)$)
    $\mathcal{W}_{opt} \leftarrow \emptyset$
    if $\mathcal{G}^c(k) \neq \emptyset$ then
        $\mathcal{T} \leftarrow$TREEOFPERMITTEDPACKAGES($\mathcal{G}^c(k)$)
        $\cup_{\mathcal{C}_j \in \mathcal{P}(k)} \bar{b}(k, \mathcal{C}_j) \leftarrow$BIDS($[\mathcal{B}_n(k)]_{n=1}^N$, $\mathcal{T}$)
        $\mathcal{W}_{opt} \leftarrow$OPTIMALALLOCATION($\mathcal{T}$, $\cup_{\mathcal{C}_j \in \mathcal{P}(k)} \bar{b}(k, \mathcal{C}_j)$)
    end if
    return $\mathcal{W}_{opt}$
end function
```



(a) Vehicle V_1 is overtaking vehicle V_2. Vehicle V_1 requests cells $\{g_0, \dots, g_{10}\}$, vehicle V_3 requests cells $\{g_0, \dots, g_8\}$, and vehicle V_2 requests cells $\{g_9, g_{10}\}$.

(b) Possible tree structure using the decomposition strategy as explained in Sec. IV-B.1.

Fig. 2. Exemplary grouping of conflicting cells according to Sec. IV-B.1.

problem (1) substantially. We further motivate the hierarchical structure of admissible packages through the following observations: we are interested in negotiating connected regions on the road surface to keep the driving area of vehicles from becoming disjointed, which complicates trajectory planning. Moreover, vehicles have to obey traffic regulations and restrictions imposed by the road network; thus, it is plausible to cluster conflicting cells according to lanes. Since using a tree structure facilitates the optimal allocation of the offered packages in $O(|\mathcal{G}^c(k)|^2)$ time complexity, it is possible to partition the road surface fine-granularly and split packages such that they solely contain single assets. The restriction that each level of the tree contains only disjoint packages $\mathcal{C}_j$ can lead to a mismatch in offered packages and desired packages, but overall, the aforementioned advantages of the hierarchical structuring of admissible packages outweigh a potential mismatch in offered and desired packages. We therefore recommend applying the strategy below to group conflicting cells (see Fig. 2; the numbering below coincides with the legend in Fig. 2b):

1) *Root*: The root node consists of all conflicting cells $\mathcal{G}^c(k)$.
2) *Connected Components:* The connected components are aggregated into packages $\mathcal{C}_j$.
3) *Road Network:* Packages of cells $\mathcal{C}_j$ are grouped according to the lanes of the road network. If a cell cannot be uniquely assigned to a lane, we categorize the cell randomly (see Fig. 2a, cell g_8).
4) *Longitudinal Spatial Coverage:* The packages are decomposed in longitudinal direction so that each new package does not exceed a maximum longitudinal spatial coverage.
5) *Lateral Spatial Coverage:* The packages are decomposed in lateral direction so that each new package does not exceed a maximum lateral spatial coverage.
6) *Singletons:* The packages comprise only singletons.

It should be noted that not all steps 1)–6) have to be executed, e.g., it is possible to apply 1) and 3) only such that the root node is split according to the road network.

2) Bids: We do not pose any specific constraint on the utility function to determine the bids $b_n(k, \mathcal{C}_j)$ for each package $\mathcal{C}_j \in \mathcal{P}(k)$ and vehicle V_n at time step k. Moreover, complementaries ($b(k, \{g_i, g_l\}) > b(k, \{g_i\}) + b(k, \{g_l\})$ for $i \neq l$) and substitutes [21] ($b(k, \{g_i, g_l\}) < b(k, \{g_i\}) + b(k, \{g_l\})$ for $i \neq l$) can be modeled. We use a common utility function to all vehicles in this paper to avoid that one vehicle could continuously outbid others due to different scales and weights used to calculate $b_n(k, \mathcal{C}_j)$. A conceivable countermeasure is the introduction of a pricing mechanism, which is the subject of future research. Furthermore, a vehicle V_n can only bid on a package $\mathcal{C}_j$ iff it contains at least one reachable cell of V_n: $\exists g_i \in \mathcal{C}_j : \; g_i \in h_n(\mathcal{B}_n(k))$.

3) Optimal Allocation: The algorithm for finding the optimal allocation of goods $\mathcal{C}_j \in \mathcal{P}(k)$ proposed by Rothkopf et al. [20] is recapitulated in Alg. 3: starting from the deepest leaf $\mathcal{C}_{\texttt{max}}$ in the tree (Alg. 3, line 6), we determine its parent node $\mathcal{C}_{\texttt{parent}}$ (Alg. 3, line 7) and the children $\mathcal{S}$ of $\mathcal{C}_{\texttt{parent}}$ (Alg. 3, line 8). Next, we compare the revenue $\texttt{rev}(\mathcal{S}) = \sum_{\mathcal{C}_s \in \mathcal{S}} \bar{b}(k, \mathcal{C}_s)$ of all children of $\mathcal{C}_{\texttt{parent}}$ with bid $\bar{b}(k, \mathcal{C}_{\texttt{parent}})$ (Alg. 3, lines 10-15):

- if $\bar{b}(k, \mathcal{C}_{\texttt{parent}}) > \texttt{rev}(\mathcal{S})$ holds, sets $\mathcal{C}_s \in \mathcal{S}$ cannot be part of the optimal allocation (Alg. 3, lines 10 - 12).
- if $\bar{b}(k, \mathcal{C}_{\texttt{parent}}) \leq \texttt{rev}(\mathcal{S})$ holds, $\mathcal{C}_{\texttt{parent}}$ is excluded from the optimal assignment (Alg. 3, lines 13 - 15).

The children $\mathcal{S}$ are removed from the tree $\mathcal{T}$ (Alg. 3, line 5), and the process is repeated until $\mathcal{C}_{\texttt{parent}}$ becomes the root node (Alg. 3, line 16): $\mathcal{C}_{\texttt{parent}} = \mathcal{G}^c(k)$.

C. Multiple Runs for Refinement

The negotiated driving corridors can be improved through multiple runs of Alg. 1. The reachable set computation is

Algorithm 3 Optimal Allocation of Packages [20].

```
function OptimalAllocation($\mathcal{T}$, $\cup_{\mathcal{C}_j \in \mathcal{P}(k)} \bar{b}(k, \mathcal{C}_j)$)
    $\mathcal{T}$.Initialize( )        ▷ Set $\mathcal{W}_{\texttt{opt}}(\mathcal{C}_l) = \{\mathcal{C}_l\}$ for every leaf $\mathcal{C}_l$.
    $\mathcal{S} \leftarrow \emptyset$
    do
        $\mathcal{T}$.RemoveNodes($\mathcal{S}$)
        $\mathcal{C}_{\texttt{max}} \leftarrow \mathcal{T}$.GetDeepestLeaf( )
        $\mathcal{C}_{\texttt{parent}} \leftarrow \mathcal{C}_{\texttt{max}}$.GetParent( )
        $\mathcal{S} \leftarrow \mathcal{C}_{\texttt{parent}}$.GetChildren( )
        $\texttt{rev}(\mathcal{S}) \leftarrow \sum_{\mathcal{C}_s \in \mathcal{S}} \bar{b}(k, \mathcal{C}_s)$
        if $\bar{b}(k, \mathcal{C}_{\texttt{parent}}) > \texttt{rev}(\mathcal{S})$ then
            $\mathcal{W}_{\texttt{opt}}(\mathcal{C}_{\texttt{parent}}) \leftarrow \{\mathcal{C}_{\texttt{parent}}\}$
        else
            $\bar{b}(k, \mathcal{C}_{\texttt{parent}}) \leftarrow \texttt{rev}(\mathcal{S})$
            $\mathcal{W}_{\texttt{opt}}(\mathcal{C}_{\texttt{parent}}) \leftarrow \cup_{\mathcal{C}_s \in \mathcal{S}} \mathcal{W}_{\texttt{opt}}(\mathcal{C}_s)$
        end if
    while $\mathcal{C}_{\texttt{parent}} \neq \mathcal{G}^{\texttt{C}}(k)$
    return $\mathcal{W}_{\texttt{opt}}(\mathcal{G}^{\texttt{C}}(k))$
end function
```

based on the results of the previous time step only. Thus, there might exist states in $\mathcal{B}_n^{\texttt{N}}(k)$ from which a trajectory cannot be continued without leaving the negotiated driving corridor $\mathcal{B}_n^{\texttt{N}}(i)$ in later time steps $i \in \{k+1, \ldots, T\}$ [23]. We are able to remove a subset of those states by running Alg. 1 multiple times, since information about future time steps from a previous run can be incorporated into the reachable set computation. The interested reader is referred to [23] for further information.

V. Utility Function and Tie-Breaking Rule

This section introduces the applied utility function (see Sec. IV-B.2) and tie-breaking rule (see Sec. II) used in this paper. Please note that both the utility function and tie-breaking rule can be exchanged by other rules.

A. Utility Function

Let us introduce:

- the conflict-free reachable set: $\mathcal{R}_n^{\texttt{CF}}(k) = \{x_n(k) \in \mathcal{B}_n(k) | h_n(\{x_n(k)\}) \cap \mathcal{G}^{\texttt{C}}(k) = \emptyset\}$;
- the conflicting reachable set depending on package $\mathcal{C}_j$ that would be lost if package $\mathcal{C}_j$ is not assigned to vehicle V_n: $\mathcal{R}_n^{\texttt{C}}(k, \mathcal{C}_j) = \{x_n(k) \in \mathcal{B}_n(k) | h_n(\{x_n(k)\}) \cap \mathcal{C}_j \neq \emptyset\}$;
- the assigned reachable set which V_n can keep besides $\mathcal{R}_n^{\texttt{CF}}(k)$ given that package $\mathcal{C}_j$ is assigned to vehicle V_n: $\mathcal{R}_n^{\texttt{A}}(k, \mathcal{C}_j) = \{x_n(k) \in \mathcal{B}_n(k) \setminus \mathcal{R}_n^{\texttt{CF}}(k) | h_n(\{x_n(k)\}) \cap (\mathcal{G}^{\texttt{C}}(k) \setminus \mathcal{C}_j) = \emptyset\}$.

The above sets are the basis for computing the utility of $\mathcal{C}_j$ for each vehicle V_n to determine $b_n(k, \mathcal{C}_j)$. For computational reasons, we approximate sets $\mathcal{R}_n^{\texttt{CF}}(k)$, $\mathcal{R}_n^{\texttt{C}}(k, \mathcal{C}_j)$, and $\mathcal{R}_n^{\texttt{A}}(k, \mathcal{C}_j)$ with the union of base sets (see Sec. IV-A.2); the approximated sets are denoted with $\mathcal{B}_n^{\texttt{CF}}(k) := \cup_i \mathcal{B}_n^{\texttt{CF}(i)}(k)$, $\mathcal{B}_n^{\texttt{C}}(k, \mathcal{C}_j) := \cup_i \mathcal{B}_n^{\texttt{C}(i)}(k)$, and $\mathcal{B}_n^{\texttt{A}}(k, \mathcal{C}_j) := \cup_i \mathcal{B}_n^{\texttt{A}(i)}(k)$, respectively. On the one hand, we take the objectives of cooperative vehicles V_n into account by applying utility function $U_n^{\texttt{R}}(k, \mathcal{C}_j)$ in the regular mode; on the other hand, we introduce utility function $U_n^{\texttt{S}}(k, \mathcal{C}_j)$ to prevent the complete loss of the reachable set $\mathcal{B}_n(k)$ of vehicles V_n in the survival mode, since this would correspond to an empty driving corridor for trajectory planning:

$$b_n(k, \mathcal{C}_j) = \begin{cases} U_n^{\texttt{R}}(k, \mathcal{C}_j), & \texttt{area}(\mathcal{B}_n^{\texttt{CF}}(k)) > \underline{A}, \\ U_n^{\texttt{S}}(k, \mathcal{C}_j), & \texttt{area}(\mathcal{B}_n^{\texttt{CF}}(k)) \leq \underline{A}, \end{cases} \tag{7}$$

where $\underline{A}$ is an adjustable threshold and $\texttt{area}(\square)$ returns the size of the drivable area of sets $\square$. As a reminder, the reachable set projected onto the position domain is referred to as the drivable area (see Sec. IV-A.2). Below, we elaborate $U_n^{\texttt{R}}(k, \mathcal{C}_j)$ and $U_n^{\texttt{S}}(k, \mathcal{C}_j)$ applied in the regular and survival mode, respectively.

1) Regular Mode: If the conflict-free drivable area of vehicle V_n is greater than $\underline{A}$, we apply $U_n^{\texttt{R}}(k, \mathcal{C}_j)$, which computes the ratio of the utility of the reachable set $\mathcal{B}_n^{\texttt{A}}(k, \mathcal{C}_j)$ obtained through package $\mathcal{C}_j$ and the utility of the conflict-free reachable set $\mathcal{B}_n^{\texttt{CF}}(k)$:

$$U_n^{\texttt{R}} = \frac{\sum_i \left(u_{\texttt{vel}}(\mathcal{B}_n^{\texttt{A}(i)}(k)) + u_{\texttt{range}}(\mathcal{B}_n^{\texttt{A}(i)}(k)) \right) \cdot \texttt{area}\left(\mathcal{B}_n^{\texttt{A}(i)}(k) \right)}{\sum_i \left(u_{\texttt{vel}}(\mathcal{B}_n^{\texttt{CF}(i)}(k)) + u_{\texttt{range}}(\mathcal{B}_n^{\texttt{CF}(i)}(k)) \right) \cdot \texttt{area}\left(\mathcal{B}_n^{\texttt{CF}(i)}(k) \right)},$$

with partial utility functions $u_{\texttt{vel}}$ and $u_{\texttt{range}}$ presented next.

In order to increase traffic flow, we reward an increase in longitudinal speed from the previous time step $k-1$ to the current time step k:

$$u_{\texttt{vel}}(\mathcal{B}^{(i)}) = y \left(\frac{\texttt{vmax}_{\zeta}(\mathcal{B}^{(i)}) - \texttt{vmax}_{\zeta}(\mathcal{B}_n^{\texttt{N}}(k-1))}{\overline{a}_{n,\zeta_n} \cdot \Delta t} \right),$$

where $\texttt{vmax}_{\zeta}(\square)$ returns the maximum velocity in ζ_n-direction of sets $\square$. We use the generalized logistic function y to scale the utility between $(0, 1]$.

Furthermore, we evaluate the covered distance in longitudinal direction from time step $k-1$ to k, since the cooperative vehicles should move forward along their reference path:

$$u_{\texttt{range}}(\mathcal{B}^{(i)}) = y \left(\frac{\texttt{pmax}_{\zeta}(\mathcal{B}^{(i)}) - \texttt{pmax}_{\zeta}(\mathcal{B}_n^{\texttt{N}}(k-1))}{\overline{v}_{n,\zeta_n} \cdot \Delta t + \frac{1}{2} \cdot \overline{a}_{n,\zeta_n} \cdot \Delta t^2} \right),$$

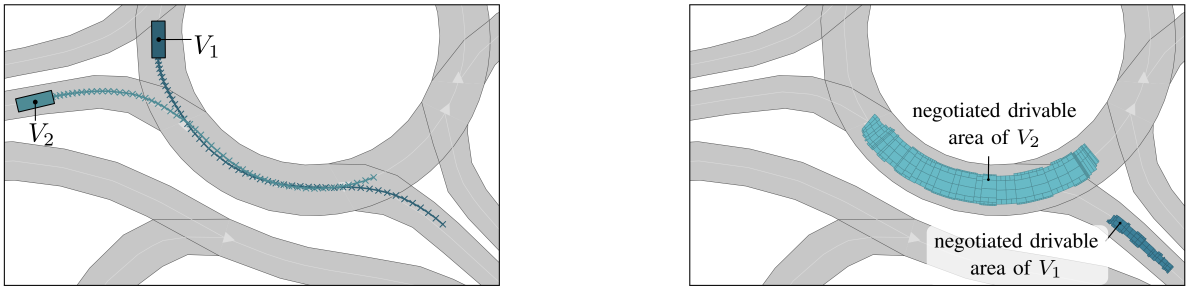
where $\texttt{pmax}_{\zeta}(\square)$ returns the maximum longitudinal position of sets $\square$. Again, we use the generalized logistic function y to scale the utility between $(0, 1]$.

2) Survival Mode: We introduce two countermeasures to avoid that the reachable sets of vehicles vanish at a certain time instance: 1) if vehicle V_n has a reachable cell $g_i \in \mathcal{C}_j$ and $\texttt{area}(\mathcal{B}_n^{\texttt{CF}}(k)) \leq \underline{A}$, no other vehicle V_m with $\texttt{area}(\mathcal{B}_m^{\texttt{CF}}(k)) > \underline{A}$ is allowed to bid on package $\mathcal{C}_j$; 2) we switch the utility function as shown in (7) to:

$$U_n^{\texttt{S}}(k, \mathcal{C}_j) = \frac{\texttt{area}(\mathcal{B}_n^{\texttt{C}}(k, \mathcal{C}_j))}{\texttt{area}(\mathcal{B}_n(k))}.$$

B. Tie-Breaking

Tie-breaking must be performed when multiple vehicles V_n bid $b_n(k, \mathcal{C}_j) = \bar{b}(k, \mathcal{C}_j)$, since this means that several optimal allocations of package $\mathcal{C}_j$ exist. In this paper, we accept the bid of the vehicle with the largest conflicting drivable area $\texttt{area}(\mathcal{B}_n(k) \setminus \mathcal{B}_n^{\texttt{CF}}(k))$; if there is a tie again,


V_1
V_2
negotiated drivable area of V_2
negotiated drivable area of V_1

(a) Vehicles V_1 and V_2 at time step $k = 0$ and their planned trajectories within the negotiated road areas of the second run.

(b) Negotiated drivable areas of vehicles V_1 and V_2 at time step $k = 55$ (first run).

Fig. 3. Scenario I: Roundabout. The driving direction is indicated by the white arrows.

TABLE I
PARAMETERS FOR NUMERICAL EXPERIMENTS.

parameter		**scenario identifier**			
symbol	**unit**	I	II	III	IV
Δt	[s]	0.1	0.1	0.1	0.1
T	/	55	50	45	34
$\overline{v}_{n,\zeta_n}$	[m/s]	15.0	28.0	13.0	18.0
$\underline{v}_{n,\zeta_n}$	[m/s]	4.0	0.0	0.0	0.0
$\overline{v}_{n,\eta_n}$	[m/s]	4.0	6.0	6.0	4.0
$\underline{v}_{n,\eta_n}$	[m/s]	-4.0	-6.0	-6.0	-4.0
$\overline{a}_{n,\zeta_n}$	[m/s^2]	2.5	4.0	4.5	2.0
$\overline{a}_{n,\eta_n}$	[m/s^2]	3.0	6.0	4.5	6.5
$\underline{A}$	[m^2]	0.0	0.0	0.0	0.0
r_n	[m]	1.3	1.2	1.2	1.3

we select the bid randomly.

VI. Evaluation

We demonstrate the universal applicability of our algorithm on four different scenarios. The selected parameters for each scenario can be found in Tab. I and are similar for all cooperative vehicles involved in a traffic scene. Please note that we only depict the drivable area with reference to the vehicle's center of gravity in the following figures to illustrate the available solution space for trajectory planning. To demonstrate that the negotiated road areas can be used for multi-vehicle trajectory planning, we show our first results for each scenario. However, trajectory planning in reachable sets is an ongoing research project and out of scope for this paper.

A. Scenario I: Roundabout

We start with the deliberately simple scenario C-DEU_B471-2_1:2018a from the CommonRoad[1] benchmark collection [24], where two communicating vehicles V_1 and V_2 cooperate such that vehicle V_2 can safely enter the roundabout (see Fig. 3a). Vehicle V_1 plans to take the first exit, while vehicle V_2 aims to take the second exit. A regular grid with tile size 0.5m×0.5m in the Cartesian reference frame F_0 is employed.

Fig. 4 shows the projected reachable sets $\mathcal{B}_n(18)$ of both vehicles and the corresponding conflicting grid cells $\mathcal{G}^{\mathrm{C}}(18)$ at time step $k = 18$ (left) and illustrates the projected negotiated reachable sets $\mathcal{B}_n^{\mathrm{N}}(18)$ (right). Vehicle V_1 accelerates and passes vehicle V_2 which enters the roundabout afterwards. The negotiated drivable areas of the final time step are depicted in Fig. 3b.

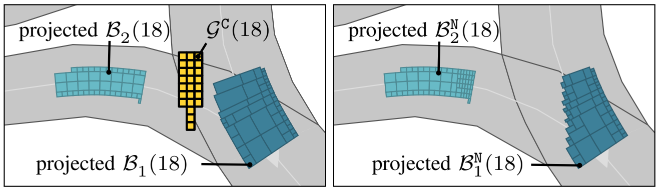

projected $\mathcal{B}_2(18)$
$\mathcal{G}^{\mathrm{C}}(18)$
projected $\mathcal{B}_1(18)$
projected $\mathcal{B}_2^{\mathrm{N}}(18)$
projected $\mathcal{B}_1^{\mathrm{N}}(18)$

Fig. 4. (Left) Projected reachable sets $\mathcal{B}_n(18)$ of V_1 and V_2 and conflicting cells $\mathcal{G}^{\mathrm{C}}(18)$ at time step $k = 18$; (Right) Projected negotiated reachable sets $\mathcal{B}_n^{\mathrm{N}}(18)$.

B. Scenario II: Urban Road

In this scenario (CommonRoad-ID: C-DEU_B471-1_1_T-1:2018a), vehicle V_1 cooperates with the oncoming vehicle V_2 to evade the static obstacle in its lane (see Fig. 5). In Fig. 5, we illustrate the result of the first and second run of the algorithm: vehicle V_1 swerves as soon as vehicle V_2 passes. At time step $k = 12$, we are able to improve the negotiated drivable area by running Alg. 1 a second time, since states of the driving corridor of vehicle V_1 leading to a collision with the static obstacle or with vehicle V_2 in a future time step are removed. Additionally, Fig. 5 visualizes the planned trajectories of vehicles V_1 and V_2 using the negotiated drivable areas of the second run.

C. Scenario III: Crossing

Following the idea of Dresner et al. [4], we allow autonomous vehicles to enter an intersection whenever it is possible. As can be seen in Fig. 6, there are two cooperating vehicles V_1 and V_2; vehicle V_1 intends to turn left, while vehicle V_2 plans to move straight ahead. We show the results after the first run of Alg. 1 in Fig. 6 in the left column. Two driving strategies occur for vehicle V_2: V_2 can either pass through the intersection before vehicle V_1 (see Fig. 6, left column, $k = 30$) or it can stop and wait until V_1 has the left the intersection (see Fig. 6, left column, $k = 45$). In this scenario, it becomes apparent that our set-based method does not only detect different cooperative maneuver options, but

[1] http://commonroad.in.tum.de

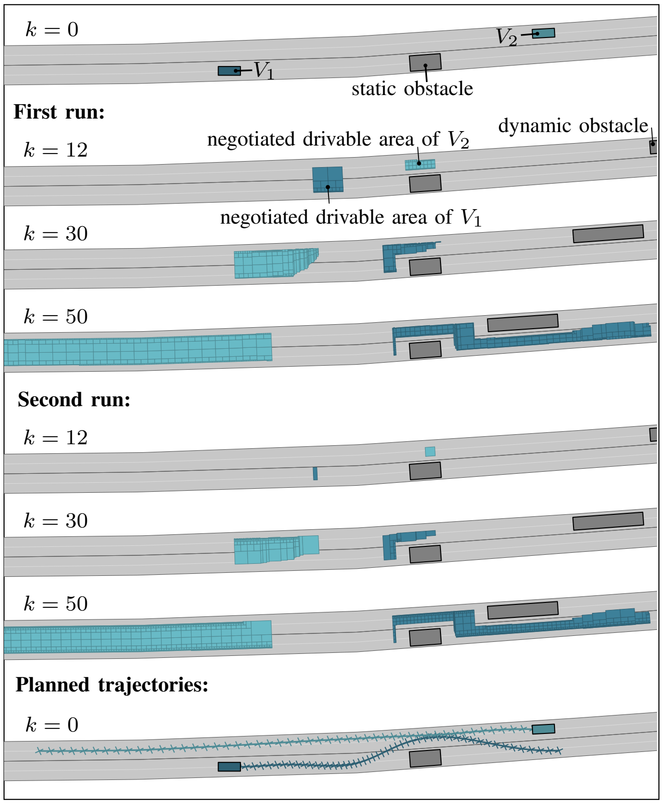


Fig. 5. Scenario II: Urban Road.

also facilitates selecting high-level plans. Driving corridors without bottlenecks are preferred, since they are more robust in terms of unpredicted changes in the environment. In this scenario, crossing the intersection can irritate human drivers and may lead to a collision, since the negotiated driving corridor for this maneuver becomes temporally tight. As illustrated in Fig. 6, the second maneuver variant—stopping at the intersection—is preferred. We therefore restrict the reachable set of V_2 to stopping at the intersection for the second run of Alg. 1 (see Sec. IV-C). However, we do not incorporate information from the first run of Alg. 1 for the reachable set computation of vehicle V_1 during the second run in order to fully utilize the released space of vehicle V_2 (see Fig. 6, right column).

D. Scenario IV: Highway

We apply our algorithm on the mixed-traffic scenario C-NGSIM_US101_1:2017a from the CommonRoad benchmark collection. We coordinate the motion of four cooperating vehicles and restrict their movement to five lanes excluding the highway on-ramp. As can be seen, our method is able to allocate road areas for cooperative vehicles in challenging traffic situations with many non-communicating traffic participants.

VII. Conclusion

We present an approach for negotiating road areas requested by multiple vehicles to determine individual driving corridors for these vehicles. The optimal allocation of offered packages can be performed with polynomial runtime complexity. Since available combinations of assets are matched to the vehicles valuing them the most, the conflict resolution is transparent. This is particularly important when considering legal issues that may arise if sub-optimality is introduced. Future research will focus on the evaluation of different utility functions to determine the bids of the cooperative vehicles.

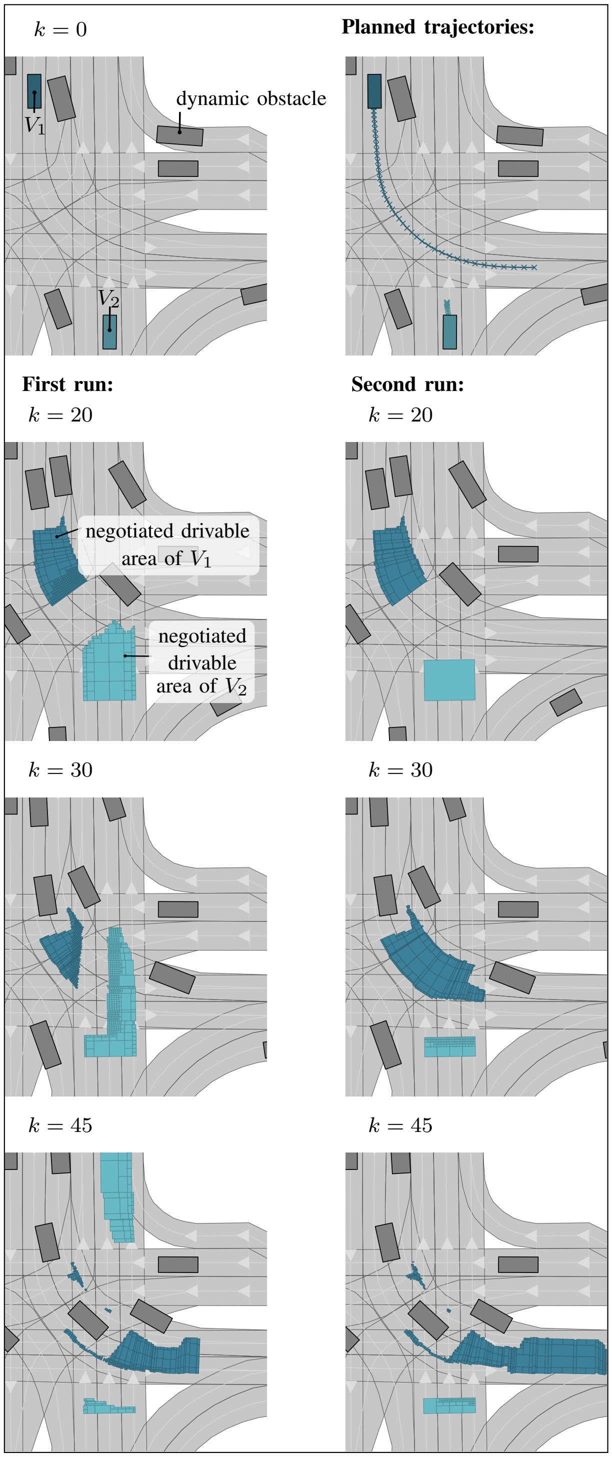


Fig. 6. Scenario III: Crossing. The driving direction is indicated by the white arrows.

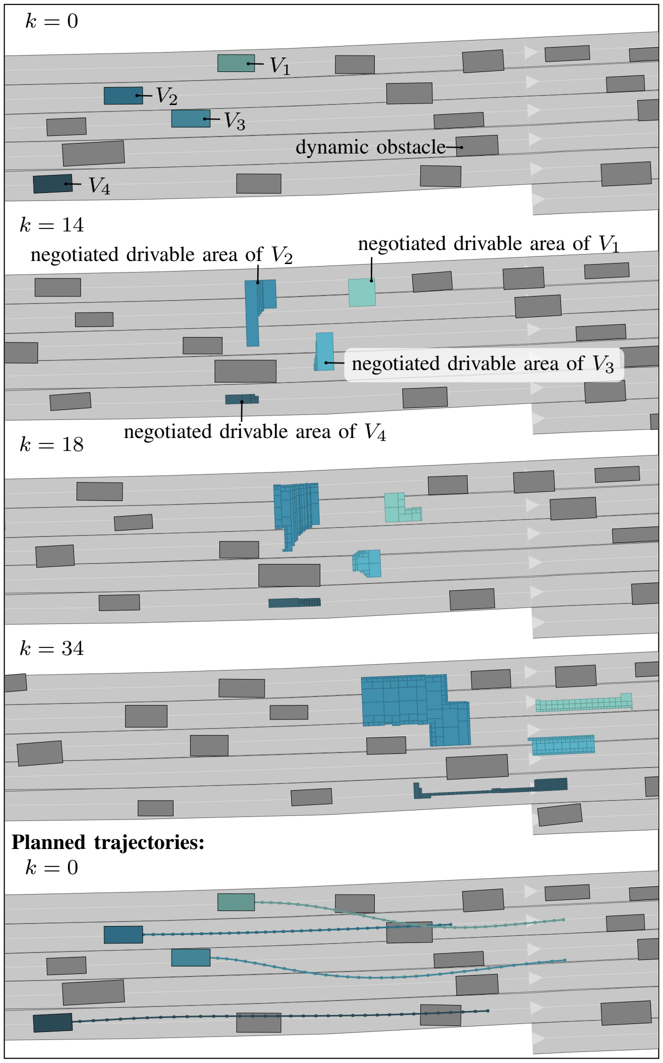

$k = 0$
V_1
V_2
V_3
dynamic obstacle
V_4
$k = 14$
negotiated drivable area of V_2
negotiated drivable area of V_1
negotiated drivable area of V_3
negotiated drivable area of V_4
$k = 18$
$k = 34$
Planned trajectories:
$k = 0$

Fig. 7. Scenario IV: Highway. The driving direction is indicated by the white arrows.

Furthermore, we plan to develop a pricing mechanism for compensating vehicles handing over driving areas.

References

[1] K. Dresner and P. Stone, "Multiagent traffic management: a reservation-based intersection control mechanism," in *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, 2004, pp. 530–537.

[2] ——, "Turning the corner: improved intersection control for autonomous vehicles," in *IEEE Intelligent Vehicles Symposium*, 2005, pp. 423–428.

[3] ——, "Multiagent traffic management: an improved intersection control mechanism," in *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems*, 2005, pp. 471–477.

[4] ——, "Human-usable and emergency vehicle-aware control policies for autonomous intersection management," in *The 4th Workshop on Agents in Traffic and Transportation*, 2006, pp. 17–25.

[5] G. Sharon and P. Stone, "A protocol for mixed autonomous and human-operated vehicles at intersections," in *Autonomous Agents and Multiagent Systems*, 2017, pp. 151–167.

[6] 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.

[7] 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.

[8] 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 5th IEEE Conference on Enterprise Computing, E-Commerce and E-Services*, 2008, pp. 105–112.

[9] F. Zhu and S. V. Ukkusuri, "A linear programming formulation for autonomous intersection control within a dynamic traffic assignment and connected vehicle environment," *Transportation Research Part C: Emerging Technologies*, vol. 55, pp. 363 – 378, 2015.

[10] 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.

[11] M. W. Levin, H. Fritz, and S. D. Boyles, "On optimizing reservation-based intersection controls," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 3, pp. 505–515, 2017.

[12] L. C. Bento, R. Parafita, S. Santos, and U. Nunes, "Intelligent traffic management at intersections: Legacy mode for vehicles not equipped with V2V and V2I communications," in *16th International IEEE Conference on Intelligent Transportation Systems*, 2013, pp. 726–731.

[13] L. Chen and C. Englund, "Cooperative intersection management: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 570–586, 2016.

[14] 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.

[15] D. Marinescu, J. Čurn, M. Slot, M. Bouroche, and V. Cahill, "An active approach to guaranteed arrival times based on traffic shaping," in *13th International IEEE Conference on Intelligent Transportation Systems*, 2010, pp. 1711–1717.

[16] J. Rios-Torres and A. A. Malikopoulos, "A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1066–1077, 2017.

[17] K. Zhang, A. Yang, H. Su, A. de La Fortelle, K. Miao, and Y. Yao, "Service-oriented cooperation models and mechanisms for heterogeneous driverless vehicles at continuous static critical sections," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 7, pp. 1867–1881, 2017.

[18] K. Zhang, A. Yang, H. Su, A. de La Fortelle, and X. Wu, "Unified modeling and design of reservation-based cooperation mechanisms for intelligent vehicles," in *IEEE 19th International Conference on Intelligent Transportation Systems*, 2016, pp. 1192–1199.

[19] S. Manzinger and M. Althoff, "Negotiation of drivable areas of cooperative vehicles for conflict resolution," in *IEEE 20th International Conference on Intelligent Transportation Systems*, 2017, pp. 1388–1395.

[20] M. H. Rothkopf, A. Pekeč, and R. M. Harstad, "Computationally manageable combinational auctions," vol. 44, no. 8, pp. 1131–1147, 1998.

[21] P. Cramton, Y. Shoham, and R. Steinberg, *Combinatorial Auctions*. MIT Press, 2006.

[22] E. Héry, S. Masi, P. Xu, and P. Bonnifait, "Map-based curvilinear coordinates for autonomous vehicles," in *IEEE 20th International Conference on Intelligent Transportation Systems*, 2017, pp. 1699–1705.

[23] S. Söntges and M. Althoff, "Computing the drivable area of autonomous road vehicles in dynamic road scenes," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 6, pp. 1855–1866, 2018.

[24] M. Althoff, M. Koschi, and S. Manzinger, "CommonRoad: Composable benchmarks for motion planning on roads," in *Proc. of the IEEE Intelligent Vehicles Symposium*, 2017, pp. 719–726.