Rule-Compliant Multi-Agent Driving Corridor Generation using Reachable Sets and Combinatorial Negotiations

Tobias Mascetta, Edmond Irani Liu, and Matthias Althoff

Abstract— **Multi-agent cooperative motion planning offers the potential to improve safety and the overall traffic flow. However, many approaches for multi-agent driving do not incorporate traffic rules or do not generalize to arbitrary scenarios. To address these open problems, we propose a novel method to negotiate individual rule-compliant driving corridors and independently plan trajectories for each controlled agent in them. We incorporate predictions into the conflict negotiation process to enable decision-making over long time horizons. Our approach is applicable to arbitrary scenarios, including mixed cooperative and non-cooperative traffic participants, as demonstrated through our numerical experiments.**

I. Introduction

Human traffic participants use non-verbal communication to resolve situations with potentially conflicting albeit rule-compliant individual goals. It is obviously desirable to transfer such capabilities to automated vehicles (AVs). We present a novel approach that realizes rule-compliant collaborative behavior by negotiating rule-compliant driving corridors computed using reachability analysis.

A. Related Work

We first provide a general introduction to multi-agent planning (MAP), after which we categorize multi-agent autonomous driving (AD) into various clusters. Finally, we discuss related work on rule-compliant multi-agent driving.

There are several approaches to MAP (see surveys in [1], [2]). One thoroughly researched approach is based on optimal control [3], while another one is focused on applying machine learning techniques, such as reinforcement learning [4] and deep reinforcement learning [5], [6]. Geometric approaches constitute a third cluster of multi-agent AD, using methods that incorporate geometric reasoning and reservation-based algorithms [7]–[10]. Methods from game theory have also been applied to multi-agent driving: Several works combine level-k game theory and offline trajectory generation with reinforcement learning. Application scenarios include highways [11]–[13], roundabouts [14], and intersections [15]. The latter use case has also been handled with auctions [16], [17]. An auction-based negotiation scheme, which is applicable to arbitrary scenarios, bundles road grid cells into packages for AVs to bid for [18]. The approach in [19] uses reachability analysis and invariably safe sets to allocate safe driving corridors to different agents. One approach combines dynamic games similar to Stackelberg games and aspects of traffic rules to enable multi-agent decision-making at intersections [20].

A line of work related to multi-agent driving is scenario synthesis. Logical scenario descriptions can be combined with optimal control to achieve rule-compliant driving of all simulated traffic participants [21]. This approach was further improved by incorporating reachable sets to achieve real-time capability [22]. However, it does not consider uncontrolled traffic participants.

An adjacent field of research is traffic simulations, using tools such as SUMO [23], Carla [24], and OpenTrafficSim [25]. Traffic simulators are able to handle large amounts of agents simultaneously and incorporate a variety of traffic rules. Yet, they only provide rudimentary microscopic motion planners and physics models. Other simulators, such as [26], combine machine learning and signal temporal logic to aim for rule-compliant multi-agent traffic simulation based on a unicycle dynamics model.

B. Contributions and Structure

Our approach is based on the auction-based negotiation from [18] for arbitrary traffic scenarios. To the best of our knowledge, we provide the first approach for multi-agent rule-compliant driving corridor generation based on reachability analysis and combinatorial negotiations. Our approach can be categorized as a hybrid framework for collaborative motion planning [2], because of the decentral reachability analysis and motion planning and the centralized negotiation strategy. Our approach incorporates predictions to make more informed decisions. The novelties of this work are:

- We provide the first approach for driving corridor generation for collaborative motion planning using reachable sets and combinatorial negotiations that considers formal traffic rules in arbitrary traffic situations.
- The motion planning inside the rule-compliant driving corridors of the vehicles is decentralized and can easily be parallelized. In contrast to other approaches, motion planning in reachable sets becomes faster for critical situations [27].
- We incorporate predictions from the reachability analysis to avoid backtracking in the subsequent motion planning.

In the following section, the problem we solve is formally introduced. Sec. III introduces required preliminaries. Our novel approach is presented in Sec. IV. Finally, we evaluate our approach on numerical experiments in Sec. V.

This work was supported by the Horizon Europe program, grant 101076165 (i4Driving).

All authors are with the School of Computation, Information and Technology at the Technical University of Munich, 85748 Garching, Germany. {tobias.mascetta, edmond.irani, althoff}@tum.de

II. Problem Statement

In this section, we formalize the problem we aim to solve. Let φ be a traffic rule encoded as a formula in linear temporal logic over finite traces *LTL_f* constructed from atomic propositions β, with the future-time connectives $\mathbf{X}$ (*next*) and $\mathbf{U}$ (*until*), according to the grammar [28]:

$$\varphi ::= \beta \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \mathbf{X}\varphi \mid \varphi_1 \mathbf{U} \varphi_2. \tag{1}$$

Let $\boldsymbol{\mathcal{G}} = \{g_0, g_1, ..., g_{m'}\}$ be a set of grid cells g_m, $m \in \{0, ..., m'\}$ of an arbitrarily shaped road network, which can be bundled into packages $\boldsymbol{\mathcal{C}}_j \in \boldsymbol{\mathcal{C}}$. Let $b_n(\boldsymbol{\mathcal{C}}_j, t)$ be the utility with which vehicle V_n values package $\boldsymbol{\mathcal{C}}_j$. Each package $\boldsymbol{\mathcal{C}}_j$ is assigned to the vehicle V_n with the highest utility $b_n(\boldsymbol{\mathcal{C}}_j, t)$, thereby setting the binary variable $\alpha_n(\boldsymbol{\mathcal{C}}_j, t) = 1$ at time t. Let $\boldsymbol{\mathcal{L}}_n^S$ define a rule-compliant set of solution trajectories τ_n for V_n such that $\forall \tau_n \in \boldsymbol{\mathcal{L}}_n^S \colon \tau_n \models \varphi$. We also introduce the operator $\textsc{Cells}(\boldsymbol{\mathcal{L}}_n^S, t)$, which returns the cells part of $\boldsymbol{\mathcal{L}}_n^S$ at time t.

We aim to find a spatio-temporal assignment of each g_m that solves the optimization problem (2) below, such that the cumulative utility of all controlled vehicles is maximized, and no cell is assigned to multiple controlled agents at the same time. Furthermore, the solution must satisfy traffic rules defined by (1). We formalize our problem as:

$$\max_{b_n(\boldsymbol{\mathcal{C}}_j, t)} \sum_{\boldsymbol{\mathcal{C}}_j \in \boldsymbol{\mathcal{C}}} \alpha_n(\boldsymbol{\mathcal{C}}_j, t)\, b_n(\boldsymbol{\mathcal{C}}_j, t) \tag{2a}$$

s.t.

$$\forall g_m \in \boldsymbol{\mathcal{G}} \colon \sum_{\{\boldsymbol{\mathcal{C}}_j \mid g_m \in \boldsymbol{\mathcal{C}}_i\}} \alpha_n(\boldsymbol{\mathcal{C}}_j, t)\, b_n(\boldsymbol{\mathcal{C}}_j, t) \leq 1 \tag{2b}$$

$$\forall \boldsymbol{\mathcal{C}}_j \in \boldsymbol{\mathcal{C}} \colon \alpha_n(\boldsymbol{\mathcal{C}}_j, t) \in \{0, 1\} \tag{2c}$$

$$\forall \boldsymbol{\mathcal{C}}_j \in \boldsymbol{\mathcal{C}} \colon \alpha_n(\boldsymbol{\mathcal{C}}_j, t) = 1 \implies \boldsymbol{\mathcal{C}}_j \subseteq \textsc{Cells}(\boldsymbol{\mathcal{L}}_n^S, t) \tag{2d}$$

In summary, we aim to compute collision-free, rule-compliant multi-agent driving corridors for motion planning.

III. Preliminaries

In this section, we introduce the formal description of rule-compliant reachability graphs and a negotiation algorithm for the combinatorial assignment of sets of goods. From hereon, we assume that all computations are applied to the n-th vehicle and we neglect the subscript n.

A. Rule-Compliant Reachable Set Computation

We define the reachable set $\mathcal{R}_k(\mathcal{X}_0)$ of a controlled vehicle V at time step k as the set of states reachable from its set of initial states $\mathcal{X}_0$ while avoiding the set of forbidden states. We compute $\mathcal{R}_k(\mathcal{X}_0)$ as in [29]. By translating an *LTL_f* formula into an equivalent non-deterministic finite automaton [28], [30], rule-compliant reachable sets $\mathcal{R}_{k,i}^S$ can be created as described in [28].

The drivable area $\mathcal{D}_k(\mathcal{R}_k(\mathcal{X}_0))$ is defined as the projection of $\mathcal{R}_k(\mathcal{X}_0)$ onto the position domain for time step k. For brevity, we use $\mathcal{D}_k$ and $\mathcal{R}_k$ from here on. Each $\mathcal{R}_k$ is split into multiple $\mathcal{R}_{k,i}$, $i \in \mathbb{N}_0$ to produce convex sets $\mathcal{D}_{k,i}$, whose union compose the drivable area $\mathcal{D}_k$ [29].

The reachability graph G stores the reachable sets $\mathcal{R}_{k,i}$ as nodes, referred to as reach nodes from hereon, and the edges are the temporal relationship between consecutive $\mathcal{R}_{k,i}$ [29]. Analogously, the rule compliant reachable sets $\mathcal{R}_{k,i}^S$ are stored in a rule-compliant reachability graph G^S. By using the rule-compliant reachability graph, we are able to incorporate information from the position and the velocity domain in the subsequent negotiation. An example reachability graph is illustrated in Fig. 3.

To employ combinatorial auctions, we create a road grid $\boldsymbol{\mathcal{G}}$, where a cell $g_{m,k}$ is only created if it intersects the drivable area of one or more vehicles at a given time step. Each $g_{m,k}$ is assigned all intersecting $\mathcal{R}_{k,i}^S$ at k. Thus, each $g_{m,k}$ can be assigned to more than one vehicle.

B. Negotiating Combinations of Road Grid Cells

The problem described in (2) is known as the *winner determination problem* and has a time complexity of $\mathcal{O}\big(2^{|\boldsymbol{\mathcal{G}}|}\big)$ [18], [31]. However, it can be solved in polynomial time with $\mathcal{O}\big(|\boldsymbol{\mathcal{G}}|^2\big)$ [31] by ex-ante defining permitted disjoint packages $\boldsymbol{\mathcal{C}}_j$ and disjoint subsets thereof.

This results in a tree structure T visualized in Fig. 2. Although the disjunctions can lead to a mismatch in supply and demand, the advantages in the computational effort outweigh this disadvantage [31], [32]. Each bidder bids for each $\boldsymbol{\mathcal{C}}_j$ and, given T, all $\boldsymbol{\mathcal{C}}_j$ are assigned by the highest bid.

Algorithm 1 $\textsc{CombinatorialAuction}$ [31]

Input: Package tree T, utility function $b(\cdot)$
Output: Optimal allocation $\boldsymbol{\mathcal{W}}_{\text{opt}}$

$\boldsymbol{\mathcal{W}}_{\text{opt}}$.INIT()
$\boldsymbol{\mathcal{C}}_{\text{max}} \leftarrow T$.GET_DEEPEST_NODE()
$\boldsymbol{\mathcal{C}}_{\text{root}} \leftarrow T$.GET_ROOT_NODE()
while $\boldsymbol{\mathcal{C}}_{\text{max}} \neq \boldsymbol{\mathcal{C}}_{\text{root}}$ **do**
  $\boldsymbol{\mathcal{C}}_{\text{siblings}} \leftarrow T$.GET_SIBLING_NODES($\boldsymbol{\mathcal{C}}_{\text{max}}$)
  $\boldsymbol{\mathcal{C}}_{\text{parent}} \leftarrow T$.GET_PARENT_NODE($\boldsymbol{\mathcal{C}}_{\text{max}}$)
  $b_{\text{siblings}} \leftarrow$ GET_CUMULATIVE_UTILITY($b(\cdot), \boldsymbol{\mathcal{C}}_{\text{siblings}}$)
  if $b(\boldsymbol{\mathcal{C}}_{\text{parent}}) > b_{\text{siblings}}$ **then**
    $\boldsymbol{\mathcal{W}}_{\text{opt}} \leftarrow \boldsymbol{\mathcal{W}}_{\text{opt}}$.EXCLUDE_ALL_SUBSETS($\boldsymbol{\mathcal{C}}_{\text{parent}}$)
    $\boldsymbol{\mathcal{W}}_{\text{opt}} \leftarrow \boldsymbol{\mathcal{W}}_{\text{opt}} \cup \{\boldsymbol{\mathcal{C}}_{\text{parent}}\}$
  else
    $\boldsymbol{\mathcal{W}}_{\text{opt}} \leftarrow \boldsymbol{\mathcal{W}}_{\text{opt}}$.PREVENT_FROM_ENTERING($\boldsymbol{\mathcal{C}}_{\text{parent}}$)
  end if
  T.REMOVE_NODES($\boldsymbol{\mathcal{C}}_{\text{siblings}}$)
  $\boldsymbol{\mathcal{C}}_{\text{max}} \leftarrow \boldsymbol{\mathcal{C}}_{\text{parent}}$
end while
return $\boldsymbol{\mathcal{W}}_{\text{opt}}$

Alg. 1 facilitates an optimal simultaneous combination and assignment of goods given T. Let $\boldsymbol{\mathcal{C}}_{\text{max}}$ be the deepest package node in T, $\boldsymbol{\mathcal{C}}_{\text{parent}}$ its parent nodes and $\boldsymbol{\mathcal{C}}_{\text{siblings}}$ all sibling nodes of $\boldsymbol{\mathcal{C}}_{\text{max}}$ including itself, and let $\boldsymbol{\mathcal{C}}_{\text{root}}$ be the root node of T. Intuitively, Alg. 1 checks from the deepest package upwards whether the value of a package is higher than the cumulative value of its constituting goods, thus terminating in finite time.

IV. Rule-Compliant Multi-Agent Driving

This section describes our approach to achieve rule-compliant multi-agent driving corridor generation. An overview of our approach is shown in Fig. 1.

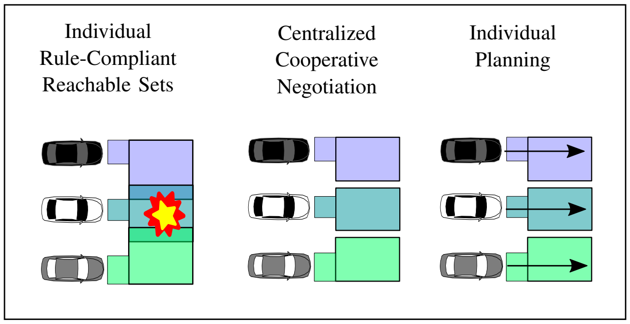


Fig. 1: Overview of our approach: We first compute the individual rule-compliant reachable sets for all controlled agents. These reachable sets are collision free with respect to uncontrolled traffic participants but they may intersect in the position domain between different controlled agents. We then perform a centralized negotiation that incorporates predictions (see Sec. IV-A to Sec. IV-C), thereby resolving potential conflicts between controlled agents. The collision-free, rule-compliant reachable sets are then used to extract rule-compliant driving corridors (see Sec. IV-D) which can be used for individual planning (see Sec. V).

A. Negotiation

We negotiate the rule-compliant drivable area by solving the *winner determination problem* using Alg. 1. To this end, we order all packages $\boldsymbol{\mathcal{C}}$ in a hierarchical tree structure as follows. All $g_{m,k}$ [18]

1) are grouped into one package;
2) are further grouped into connected sets;
3) are further grouped by their lanelets [22];
4) are further grouped in longitudinal spatial intervals;
5) are further grouped in lateral spatial intervals.

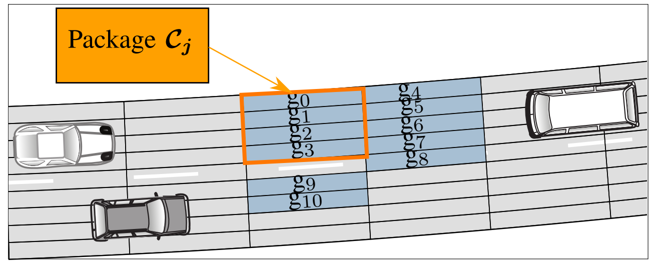
Only packages $\boldsymbol{\mathcal{C}}_j$ with at least one conflicting cell, denoted by $\boldsymbol{\mathcal{C}}_j^C$, are up for bidding, whereas conflict-free cells, denoted by $\boldsymbol{\mathcal{C}}_j^F$, can be trivially assigned to V. If a node $\mathcal{R}_{k,i}^S$ loses at least one cell $g_{m,k}$ associated with its drivable area $\mathcal{D}_{k,i}$, that node is considered lost entirely, as otherwise, the reachable set propagation as described in Sec. III-A would become invalid.

When negotiating reachable sets using Alg. 1, an important detail is that each $\mathcal{D}_{k,i}$ must not exceed a given area. An intuitive counter-example is a scenario in which two vehicles only have one $\mathcal{R}_{k,i}^S$ for all $k \in [0, k_{\text{final}}]$, which causes one agent to lose its entire $\mathcal{D}_k$ in each negotiation step. Given the threshold $\epsilon_{split} \in \mathbb{R}_{>0}$ for the maximal scalar value of the drivable area and partition sizes h_ζ and h_η respectively, each drivable area whose area is greater than ϵ_{split} is split along the curvilinear coordinates axes ζ and η into smaller drivable areas.

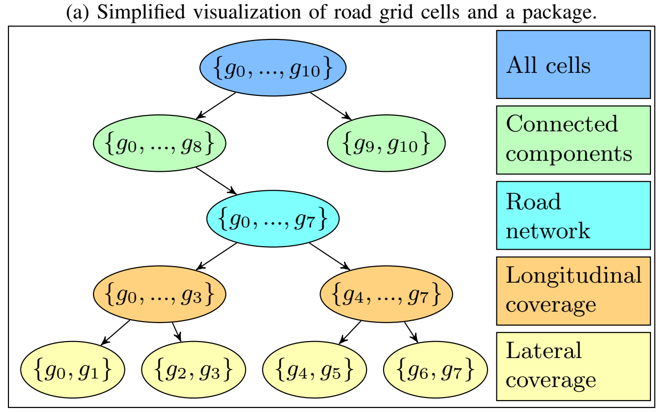
B. Using Predictions in Negotiations

We enable cooperative decision-making over multiple time steps by considering predictions in the negotiation. The concept is illustrated in Fig. 3, where the elimination of one node causes other nodes to become invalid.

In Alg. 2, we exploit that G^S is a connected, directed, acyclic graph (DAG) and apply the following graph manipulation methods for recursively deleting nodes: We first gather all nodes that belong to the conflicting package $\boldsymbol{\mathcal{C}}_j^C$ and temporarily remove them from a copy of G^S (lines 1-5). Then, we use a breadth-first search (BFS) to go backward in time in G^S and temporarily remove all $\mathcal{R}_{k,i}^S$ that would have lost all their child nodes $\mathcal{R}_{k,i}^S$, if V lost $\boldsymbol{\mathcal{C}}_j^C$ (line 6) [33]. We then go forward in time to the final time step and temporarily remove all $\mathcal{R}_{k,i}^S$ that would transitively have lost all their parent nodes $\mathcal{R}_{k,i}^S$ (line 7), which is again a BFS. Let $\rho(G^S)$ be the entire drivable area stored in G^S over the entire time horizon. Furthermore, let $\rho^l(G^S, \boldsymbol{\mathcal{C}}_j^C, k)$ be the area over the entire time horizon that would be lost if $\boldsymbol{\mathcal{C}}_j^C$ were lost at time step k, computed as the difference in area between the original graph G^S and the temporarily modified copy (line 8). Since G^S is a connected DAG and our graph traversal is based on BFS, Alg. 2 has a linear time-complexity of $\mathcal{O}\big(|\boldsymbol{\mathcal{N}}_{G^S}| + |\boldsymbol{\mathcal{E}}_{G^S}|\big)$ [33, Sec. 3.1], where $\boldsymbol{\mathcal{N}}_{G^S}$ denotes the nodes in G^S, and $\boldsymbol{\mathcal{E}}_{G^S}$ the edges, respectively.



(a) Simplified visualization of road grid cells and a package.



(b) Grouping of conflicting cells for negotiation. Note that we do not negotiate singleton-cell packages to avoid too fine-grained packages.

Fig. 2: Example of a grid-based negotiation tree with reachable sets. Adapted from [18] to our approach.

C. Utility Functions

We do not impose specific constraints on the utility function used in the negotiation and define a regular-mode utility function $U^R(\boldsymbol{\mathcal{C}}_j^C, k)$ as well as a survival-mode utility function $U^S(\boldsymbol{\mathcal{C}}_j^C, k)$ as in [18]. If a controlled agent in survival mode bids for a conflicting package $\boldsymbol{\mathcal{C}}_j^C$, only other controlled agents in survival mode can bid for it to avoid the respective $\mathcal{D}_k$ becoming empty. Let $d_k(\cdot)$ be the scalar value of the area of the drivable area $\mathcal{D}_k$ associated to a package and let $\boldsymbol{\mathcal{C}}^D$ be the set of all packages that intersect

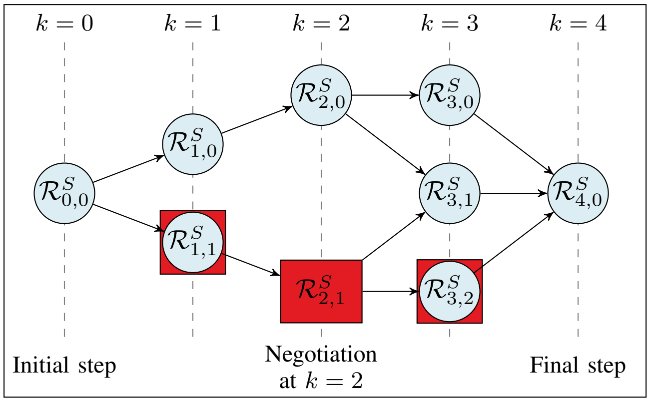


Fig. 3: Reachability graph with time steps from $k=0$ to $k=4$. At $k=2$, the red, rectangular node is in conflict for the given vehicle. Traversing the reachability graph reveals that the two blue circular nodes with red rectangular outlines would lose their only child or parent node and are thus no longer relevant.

Algorithm 2 GRAPHMANIPULATION

Input: Rule-compliant reachability graph G^S, package $\boldsymbol{\mathcal{C}}^C_j$
Output: Area lost $\rho^l(G^S, \boldsymbol{\mathcal{C}}^C_j, k)$
$G^S_{\text{temp}} \leftarrow G^S$.COPY_GRAPH()
$\boldsymbol{\mathcal{R}}^S \leftarrow \boldsymbol{\mathcal{C}}^C_j$.GET_ASSOCIATED_NODES()
for each $\mathcal{R}^S_{k,i}$ in $\boldsymbol{\mathcal{R}}^S$ **do**
  $G^S_{\text{temp}} \leftarrow G^S_{\text{temp}}$.REMOVE_NODE($\mathcal{R}^S_{k,i}$)
end for
$G^S_{\text{temp}} \leftarrow G^S_{\text{temp}}$.REMOVE_NODES_WITHOUT_CHILDREN($\mathcal{R}^S_{k,i}$)
$G^S_{\text{temp}} \leftarrow G^S_{\text{temp}}$.REMOVE_NODES_WITHOUT_PARENTS($\mathcal{R}^S_{k,i}$)
$\rho^l(G^S, \boldsymbol{\mathcal{C}}^C_j, k) \leftarrow G^S$.AREA() $-$ G^S_{temp}.AREA()
return $\rho^l(G^S, \boldsymbol{\mathcal{C}}^C_j, k)$

with the drivable area of a vehicle at time step k before the negotiation. We introduce the threshold ϵ_A defining the minimal allowed cumulative area of $\mathcal{D}_k$ before entering survival mode with the utility $U^S(\boldsymbol{\mathcal{C}}^C_j, k)$. Thus, a bid is

$$b_k(\boldsymbol{\mathcal{C}}^C_j) = \begin{cases} U^S(\boldsymbol{\mathcal{C}}^C_j, k), & \text{if } d_k(\boldsymbol{\mathcal{C}}^D) \le \epsilon_A \\ U^R(\boldsymbol{\mathcal{C}}^C_j, k), & \text{otherwise.} \end{cases} \quad (3)$$

We apply the tie-break strategy from [18] for equal bids. Let the subscripts min and max denote the minimum and maximum values in a curvilinear coordinate system for the functions $\zeta(\cdot)$, $\eta(\cdot)$, and $\dot{\zeta}(\cdot)$, which return the curvilinear longitudinal and lateral position and velocity, respectively; these bounds are directly obtained from the reachable sets. Given the logistic function $y(\cdot)$ and weights $w_{\square} \in \mathbb{R}_{>0}$, we define the following partial utility functions:

- The partial utility function
$$u_p(\boldsymbol{\mathcal{C}}^C_j, k) = w_p\, y\big(\zeta_{\max}(\boldsymbol{\mathcal{C}}^C_j) - \zeta_{\max}(x_{k-1})\big) \quad (4)$$
promotes moving forward in ζ-direction along the reference path Γ towards the goal.
- The partial utility function
$$u_v(\boldsymbol{\mathcal{C}}^C_j, k) = w_v\, y\big(\dot{\zeta}_{\max}(\boldsymbol{\mathcal{C}}^C_j) - \dot{\zeta}_{\max}(x_{k-1})\big) \quad (5)$$
promotes a higher speed in ζ-direction along Γ.
- The partial utility function
$$u_r(\boldsymbol{\mathcal{C}}^C_j, k) = w_r\, e^{-|\eta_{\min}(\boldsymbol{\mathcal{C}}^C_j)|} \quad (6)$$
promotes staying close to Γ in η-direction along Γ, thus penalizing absolute lateral divergence from Γ.
- The partial utility function
$$u_g(\boldsymbol{\mathcal{C}}^C_j, k) = w_g\, \frac{\rho^l(G^S, \boldsymbol{\mathcal{C}}^C_j, k)}{\rho(G^S)} \quad (7)$$
increases the importance of packages that have a bigger influence on the size of the solution area over the entire scenario horizon.

Together, these utility functions promote following the reference path Γ of the curvilinear coordinate system as closely as possible. Using an abbreviated notation for clarity, let us define

$$U^R(\boldsymbol{\mathcal{C}}^C_j, k) = \frac{(u_p + u_v + u_r)d_k(\boldsymbol{\mathcal{C}}^C_j) + u_g}{(u_p + u_v + u_r)d_k(\boldsymbol{\mathcal{C}}^F)} \quad (8)$$

as the combined utility, where $\boldsymbol{\mathcal{C}}^F$ is the union set of all $\boldsymbol{\mathcal{C}}^F_j$ of vehicle V at time step k. Note that this definition does not result in a division by zero, as a vanishing collision free drivable area triggers the survival mode utility from [18]. Once the packages of one negotiation iteration are assigned, we update the reachability graph G^S of all vehicles $V \in \boldsymbol{\mathcal{V}}$ in the same manner as described in IV-B.

D. Extracting Driving Corridors

From the negotiated graph G^S, we extract driving corridors $\boldsymbol{\mathcal{K}}$, defined as a set of spatially connected subsets of $\mathcal{R}^S_{k,i}$ which are also temporally connected through the edges in G^S [29]. For computational efficiency, we use the grid-based re-partitioning in [29] as a preprocessing step.

Contrary to previous works in [29], [30], our driving corridor extraction promotes moving towards the goal state even in the presence of disjoint drivable areas, a phenomenon often occurring in multi-agent AD. Let us denote a spatially connected node by Q_k and $l(Q_k)$ as the maximum longitudinal position within the drivable area of Q_k to define the longitudinal distance metric between two temporally consecutive nodes Q_k and Q_{k-1} as

$$d(Q_k, Q_{k-1}) = \min\{0, -(l(Q_k) - l(Q_{k-1}))\}. \quad (9)$$

We apply Dijkstra's Algorithm with the cost function specified in (9) to $\tilde{G}^S$ whose nodes are now spatially connected nodes Q_k. This prioritizes corridors $\boldsymbol{\mathcal{K}}$ that move towards the goal state even in the presence of disjoint drivable areas. We select the driving corridor with the lowest path costs from Dijkstra's Algorithm that intersects the goal polygon or has the smallest Hausdorff distance to it. If multiple driving corridors $\boldsymbol{\mathcal{K}}$ fulfill this condition, we choose the one with the biggest cumulative area.

V. EVALUATION

We demonstrate the cooperative rule-compliant capabilities of our approach in mixed-traffic scenarios chosen from the CommonRoad benchmark suite[1]. For our demonstration, we use a planner based on quadratic programming that treats the longitudinal and lateral planning in reachable sets consecutively [34] with a double-integrator model with bounded velocity and acceleration described in [22]. Let $\Delta x_f = x_{\text{target}} - x_{k_{\text{final}}}$, and R and Q be cost matrices, respectively and let $a \in \mathbb{R}^2$ be the vector describing the longitudinal and lateral acceleration. We use the following cost function for each controlled agent:

$$J(x_k, u_k) = \sum_{k=0}^{k_{\text{final}}-1} a^{\mathrm{T}} R a + \Delta x_f^{\mathrm{T}} Q \Delta x_f. \qquad (10)$$

Note that such a consecutive planning approach is not guaranteed to be feasible, however the fail-safe concept described in [35] can be applied.

We consider traffic rules for highway merging, intersections, and closed roads as defined in [28], [30]; further traffic rules can be implemented in LTL_f [30], [36], [37]. Tab. I contrasts our approach with the works in [18], [22].

TABLE I: QUALITATIVE COMPARISON OF OUR APPROACH TO [18] AND [22].

Comparison metrics	OURS	[18]	[22]
Framework	Hybrid	Hybrid	Hybrid
Multi-agent planning	✓	✗	✓
Mixed traffic	✓	✓	✗
Traffic rules	✓	✗	✓
Multiple time steps in negotiation	✓	✗	✗

For our numerical experiments, we choose $w_p = w_v = w_r = 1$ and $w_g = 10$, $\epsilon_A = 5$, $\epsilon_{\text{split}} = 2.5$, $h_\zeta = 4$, $h_\eta = 2$, $\Delta t = 0.1s$, and a road grid edge size of 0.2m. Moreover, we use CVXOPT[2] to solve the problem given in (10). For our illustrations, we employ the CommonRoad visualization scheme shown in Fig. 4.

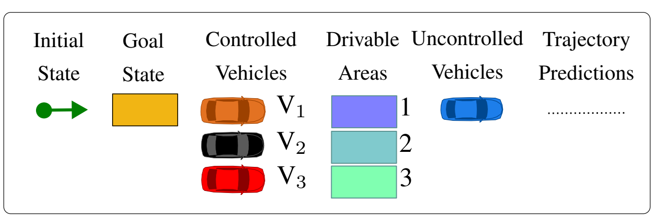


Fig. 4: CommonRoad visualization legend. The trajectory predictions are from the perspective of V_1.

A. Merging Scenarios

We start by evaluating the merging scenario `ZAM_Merge_TR_MA`. V_1 and V_2 drive in the left lane while V_3 merges into the right lane, as shown in Fig. 5. All vehicles have the same goal lanelet. Additionally, the initial velocity of V_1 is deliberately chosen higher than the velocity of V_2 to provoke a potential crash situation. Vehicles V_1 and

[1]https://commonroad.in.tum.de/scenarios
[2]https://cvxopt.org/

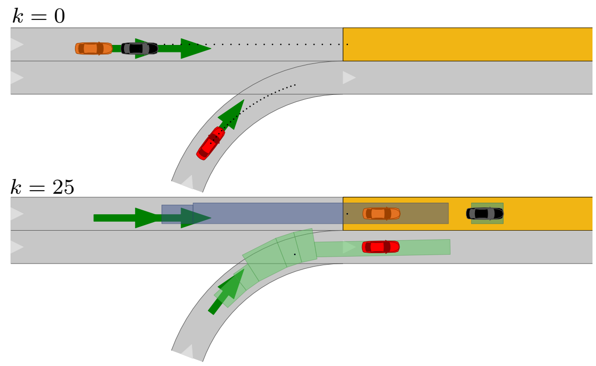


Fig. 5: Merge scenario without uncontrolled traffic participants.

V_2 comply with the merging rule by not entering the right lane. Our approach avoids the potential crash between V_1 and V_2 by accelerating V_2 and breaking V_1. Furthermore, the negotiation allows V_1 and V_2 to enter the goal lanelet before V_3 since they are faster than V_3.

Next, we evaluate the mixed-traffic capabilities of the merging rule on the real-world scenario `DEU_MerzenichRather-2_8814400_T-14549`. Here, V_1 and V_3 are in the middle lane, V_2 is in the left lane, and a humand-driven car is merging into the road. The goal state of V_1 is deliberately chosen to be close to the other controlled agents to provoke a crash situation and too small to stay in it safely. As shown in Fig. 6, the three controlled vehicles do not try to enter the right lanelet, thereby obeying the merge rule. Furthermore, since they all try to be within the goal region but cannot stay in it safely, they order themselves in a way that each intersects with the goal region at a different time step. This causes V_3 to accelerate and V_1 to decelerate, providing space to V_2. Afterwards, all three controlled vehicles move to the left lanelet again to overtake the slower uncontrolled vehicle.

B. Intersection Scenarios

We evaluate the right-before-left rule[3] on the scenario `ZAM_Intersection_Only_Agents` which only involves controlled agents. Vehicles V_1 and V_3 have to give way to V_2 coming from the right. Additionally, there is a crash potential between V_1 and V_3. As shown in Fig. 7, the right-before-left rule is obeyed and V_1 and V_3 brake while respecting a safe distance to one another.

Scenario `ZAM_Intersection-1_2_T-1` models a mixed-traffic intersection. As shown in Fig. 8, V_1 and V_2 have the same planning problem as in the scenario with only controlled agents but also comply with the right-before-left rule with respect to uncontrolled vehicles.

C. Closed Road with Road Block

We evaluate the closed-road rule on the scenario `DEU_Test-1_2_T-1`. The right lane is closed and there is an additional static road block in the middle lane. As shown

[3]In some countries, at unsigned intersections, the vehicle coming from the right has the right of way.

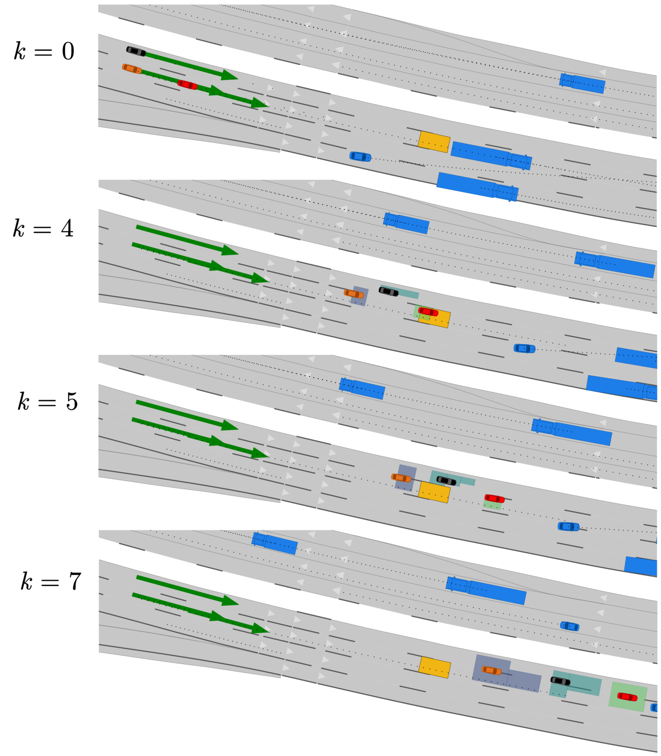


Fig. 6: Mixed-traffic merge scenario. In this scenario, we use $\Delta t = 0.4s$ due to the given data source [38].

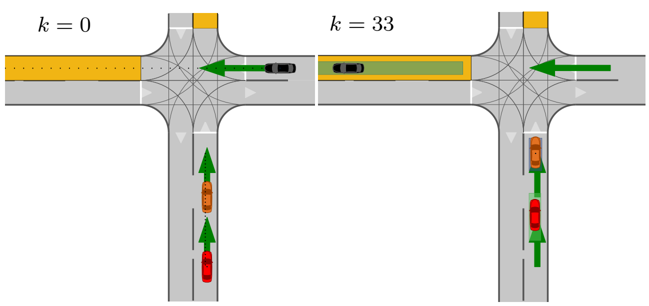


Fig. 7: Intersection scenario without uncontrolled vehicles.

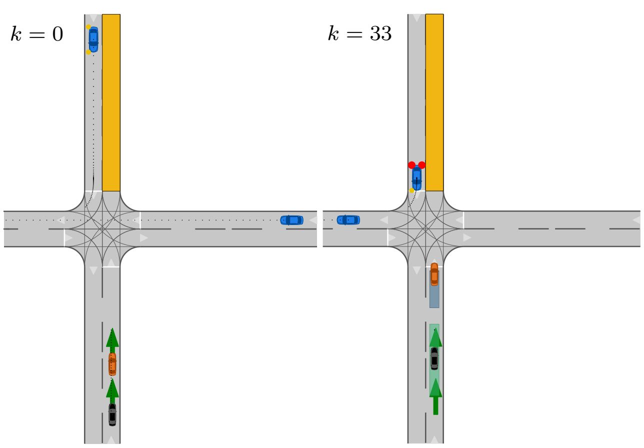


Fig. 8: Mixed-traffic intersection scenario.

in Fig. 9, both vehicles choose the open left lane to evade the obstacle. Additionally, they increase the distance between them since both have enough time and space to reach their respective goals.

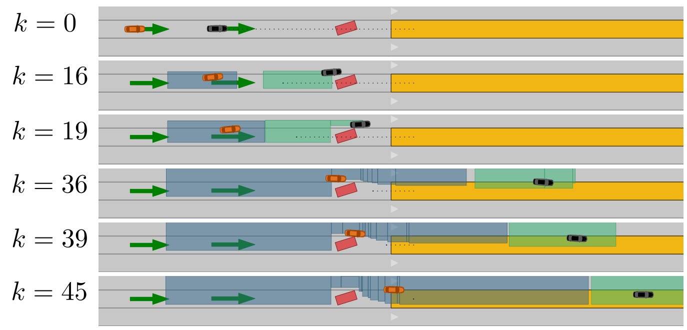


Fig. 9: Road block scenario.

D. *Dead End*

We evaluate the influence of predictions, especially in the utility function, on the scenario `C-DEU_B471-1_1_T-6`. The vehicle V_1 is deliberately placed at the entrance of a long dead end, while V_2 drives on its left with equal velocity, which obstructs the necessary lane change of V_1 in order to avoid the dead end. Each vehicle has its respective goal state in the same lane as its initial state. Without the additional term in (7) on predictions as in [18], V_1 is negotiated into the dead end, whereas with our novel approach, vehicle V_2 overtakes a decelerating vehicle V_1, thereby preventing V_1 from driving into the dead end, as shown in Fig. 10.

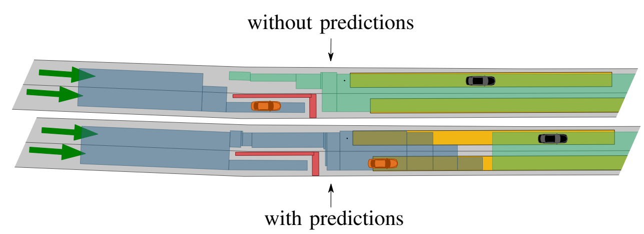


Fig. 10: The negotiation without predictions cannot avoid the dead end, in contrast to our approach. Depicted is the last time step $k = 47$ for both approaches.

VI. Conclusion

We present an approach for collaborative, rule-compliant driving corridor generation in mixed scenarios of arbitrary type. The controlled agents plan their individual trajectories in a decentralized way inside rule-compliant, collision-free driving corridors. However, our overall approach negotiates rule-compliant reachable sets over the entire time horizon in a centralized manner. Our experiments show that our approach is applicable to different scenario types and different traffic rules in mixed-traffic scenarios.

Information on the reachable sets of the entire time horizon is used to make more informed decisions compared to previous approaches. The generation of the corridors is transparent, which is important regarding legal issues. To our best knowledge, this approach is the first presenting an approach to multi-agent rule-compliant driving corridor generation by combining reachability analysis, combinatorial negotiations, as well as predictions.

REFERENCES

[1] A. Torreno, E. Onaindia, A. Komenda, and M. Štolba, "Cooperative multi-agent planning: A survey," *ACM Computing Surveys*, vol. 50, no. 6, pp. 1–32, 2017.

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

[3] R. M. Murray, "Recent research in cooperative control of multivehicle systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 129, no. 5, pp. 571–583, 2007.

[4] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "Safe, multi-agent, reinforcement learning for autonomous driving," *arXiv preprint arXiv:1610.03295*, 2016.

[5] A. Sharif and D. Marijan, "Evaluating the robustness of deep reinforcement learning for autonomous policies in a multi-agent urban driving environment," in *22nd IEEE International Conference on Software Quality, Reliability and Security*, 2022, pp. 785–796.

[6] P. Palanisamy, "Multi-agent connected autonomous driving using deep reinforcement learning," in *IEEE International Joint Conference on Neural Networks*, 2020, pp. 1–7.

[7] K. Dresner and P. Stone, "Multiagent traffic management: A reservation-based intersection control mechanism," in *International Joint Conference on Autonomous Agents and Multiagent Systems*, vol. 3, 2004, pp. 530–537.

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

[9] D. Marinescu, J. Čurn, M. Bouroche, and V. Cahill, "On-ramp traffic merging using cooperative intelligent vehicles: A slot-based approach," in *15th IEEE International Conference on Intelligent Transportation Systems*, 2012, pp. 900–906.

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

[11] D. W. Oyler, Y. Yildiz, A. R. Girard, N. I. Li, and I. V. Kolmanovsky, "A game theoretical model of traffic with multiple interacting drivers for use in autonomous vehicle development," in *American Control Conference*, 2016, pp. 1705–1710.

[12] N. Li, D. W. Oyler, M. Zhang, Y. Yildiz, A. Girard, and I. V. Kolmanovsky, "Hierarchical reasoning game theory based approach for evaluation and testing of autonomous vehicle control systems," in *55th IEEE Conference on Decision and Control*, 2016, pp. 727–733.

[13] N. Li, D. W. Oyler, M. Zhang, Y. Yildiz, I. Kolmanovsky, and A. R. Girard, "Game theoretic modeling of driver and vehicle interactions for verification and validation of autonomous vehicle control systems," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 5, pp. 1782–1797, 2017.

[14] R. Tian, S. Li, N. Li, I. Kolmanovsky, A. Girard, and Y. Yildiz, "Adaptive game-theoretic decision making for autonomous vehicle control at roundabouts," in *57th IEEE Conference on Decision and Control*, 2018, pp. 321–326.

[15] N. Li, I. Kolmanovsky, A. Girard, and Y. Yildiz, "Game theoretic modeling of vehicle interactions at unsignalized intersections and application to autonomous vehicle control," in *American Control Conference*, 2018, pp. 3215–3220.

[16] 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.

[17] D. Carlino, S. D. Boyles, and P. Stone, "Auction-based autonomous intersection management," in *16th IEEE International Conference on Intelligent Transportation Systems*, 2013, pp. 529–534.

[18] S. Manzinger and M. Althoff, "Tactical decision making for cooperative vehicles using reachable sets," in *21st IEEE International Conference on Intelligent Transportation Systems*, 2018, pp. 444–451.

[19] E. Irani Liu, C. Pek, and M. Althoff, "Provably-safe cooperative driving via invariably safe sets," in *IEEE Intelligent Vehicles Symposium*, 2020, pp. 516–523.

[20] N. Li, Y. Yao, I. V. Kolmanovsky, E. Atkins, and A. R. Girard, "Game-theoretic modeling of multi-vehicle interactions at uncontrolled intersections," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 1428–1442, 2022.

[21] M. Klischat and M. Althoff, "Synthesizing traffic scenarios from formal specifications for testing automated vehicles," in *IEEE Intelligent Vehicles Symposium*, 2020, pp. 2065–2072.

[22] F. Finkeldei and M. Althoff, "Synthesizing traffic scenarios from formal specifications using reachability analysis," in *26th IEEE International Conference on Intelligent Transportation Systems*, 2023.

[23] D. Krajzewicz, G. Hertkorn, C. Rössel, and P. Wagner, "SUMO (Simulation of Urban MObility)-an open-source traffic simulation," in *4th Middle East Symposium on Simulation and Modelling*, 2002, pp. 183–187.

[24] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *1st Conference on Robot Learning*. PMLR, 2017, pp. 1–16.

[25] H. van Lint, W. Schakel, G. Tamminga, P. Knoppers, and A. Verbraeck, "Getting the human factor into traffic flow models: New open-source design to simulate next generation of traffic operations," vol. 2561, no. 1, pp. 25–33, 2016.

[26] Z. Zhong, D. Rempe, D. Xu, Y. Chen, S. Veer, T. Che, B. Ray, and M. Pavone, "Guided conditional diffusion for controllable traffic simulation," in *IEEE International Conference on Robotics and Automation*, 2023.

[27] G. Würsching and M. Althoff, "Sampling-based optimal trajectory generation for autonomous vehicles using reachable sets," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2021, pp. 828–835.

[28] F. Lercher and M. Althoff, "Specification-compliant reachability analysis for autonomous vehicles using on-the-fly model checking," *accepted to 35th IEEE Intelligent Vehicles Symposium*, 2024.

[29] E. Irani Liu, G. Würsching, M. Klischat, and M. Althoff, "Commonroad-Reach: A toolbox for reachability analysis of automated vehicles," in *25th IEEE International Conference on Intelligent Transportation Systems*, 2022, pp. 2313–2320.

[30] E. Irani Liu and M. Althoff, "Specification-compliant driving corridors for motion planning of automated vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 9, pp. 4180–4197, 2023.

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

[32] A. Pekeč and M. H. Rothkopf, "Combinatorial auction design," *Management science*, vol. 49, no. 11, pp. 1485–1503, 2003.

[33] A. Ferenczi and C. Bădică, "An implementation of depth-first and breadth-first search algorithms for tip selection in iota distributed ledger," in *Intelligent Information and Database Systems*. Springer International Publishing, 2022, pp. 351–363.

[34] S. Manzinger, C. Pek, and M. Althoff, "Using reachable sets for trajectory planning of automated vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 2, pp. 232–248, 2020.

[35] C. Pek, S. Manzinger, M. Koschi, and M. Althoff, "Using online verification to prevent autonomous vehicles from causing accidents," *Nature Machine Intelligence*, vol. 2, no. 9, pp. 518–528, 2020.

[36] S. Maierhofer, P. Moosbrugger, and M. Althoff, "Formalization of intersection traffic rules in temporal logic," in *IEEE Intelligent Vehicles Symposium*, 2022, pp. 1135–1144.

[37] S. Maierhofer, A.-K. Rettinger, E. C. Mayer, and M. Althoff, "Formalization of interstate traffic rules in temporal logic," in *IEEE Intelligent Vehicles Symposium*, 2020, pp. 752–759.

[38] T. Moers, L. Vater, R. Krajewski, J. Bock, A. Zlocki, and L. Eckstein, "The exid dataset: A real-world trajectory dataset of highly interactive highway scenarios in germany," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, 2022, pp. 958–964.