2016 IEEE Intelligent Vehicles Symposium (IV)
Gothenburg, Sweden, June 19-22, 2016

Tactical Cooperative Planning for Autonomous Highway Driving using Monte-Carlo Tree Search

David Lenz[1] and Tobias Kessler[1] and Alois Knoll[2]

Abstract— **Human drivers use nonverbal communication and anticipation of other drivers' actions to master conflicts occurring in everyday driving situations. Without a high penetration of vehicle-to-vehicle communication an autonomous vehicle has to have the possibility to understand intentions of others and share own intentions with the surrounding traffic participants. This paper proposes a cooperative combinatorial motion planning algorithm without the need for inter vehicle communication based on Monte Carlo Tree Search (MCTS). We motivate why MCTS is particularly suited for the autonomous driving domain. Furthermore, adoptions to the MCTS algorithm are presented as for example simultaneous decisions, the usage of the Intelligent Driver Model as microscopic traffic simulation, and a cooperative cost function. We further show simulation results of merging scenarios in highway-like situations to underline the cooperative nature of the approach.**

I. Introduction

Highly automated or autonomous vehicles are on the verge of becoming reality. Every year, new demonstrations show the technical feasibility of automatically driving on highways, rural roads or even a few vehicles mastering urban traffic. Although most of the time, the environmental conditions like weather, traffic density, etc during experiments are still controlled to allow a safe operation, this has been a huge development over the last decade.

There even already exist technical solutions in commercial vehicles starting from traffic jams up to the AutoPilot feature of Tesla. Although these features are still limited to relatively low speeds, restricted lane changing, only highways or through constant monitoring through the driver.

Almost all solution have in common, that the behavior of the automated vehicle is merely *reacting* to changes in the environment. As human drivers, we usually expect some behavior of other vehicles. In Fig. 1 for example, we might assume, that the other vehicle lets us merge onto a lane. In dense traffic, this kind of cooperation and anticipation is vital in order to successfully change a lane. On the other hand, we also want to show anticipatory cooperative behavior in such situations to let other vehicles merge without the need of communication. This should happen before a merge starts to avoid strong reactions of our vehicle to avoid collision with the incoming vehicle. More general, Ulbrich *et al.* [1] gives a brief overview over situations where cooperation is necessary and usually shown by human drivers.

[1]David Lenz and Tobias Kessler are with fortiss GmbH, affiliated institute of Technische Universität München, Munich, Germany

[2]Alois Knoll is with Robotics and Embedded Systems, Technische Universität München, Munich, Germany

Fig. 1. A typical driving scene making cooperation necessary for the vehicle on the right lane to successfully make a lane change. In dense traffic it is not possible to merge, if only predicting behavior based on past behavior. It is necessary to account for reactions when an highly automated vehicle is trying to merge.

In this paper, we therefor address the problem of generating anticipatory and cooperative behavior of the automated vehicle, that takes the influence of ones actions to the other traffic participants and the traffic flow into account.

The main contributions of this paper are

- the adoption of the well-known MCTS algorithm to the autonomous driving domain
- the presentation of a cooperative motion planning algorithm without communication between the participants
- the demonstration of the technical feasibility in different challenging lane merging scenarios.

The structure of this paper is as follows: First, we present work directly related to the presented work. Then we define the problem formally and introduce the core algorithm of this paper TCMP-MCTS. Last, we perform multiple experiments with simulations and discuss the results.

II. Related Work

In the Urban Grand Challenge, multiple solution for tactical planning were proposed, that were specifically tailored for the challenge. Montemerlo *et al.* [2] uses a finite state machine and Kammel *et al.* [3] a hybrid state machine to switch between predefined behaviors. Bacha *et al.* [4] use a behavior based high-level planning approach with many agents that are selected with an arbitration block containing finite state machines. These rule-based approaches lack the ability to generalize to unknown situations and deal with uncertainties in the inputs.

Damerow and Eggert [5] give a planning framework based on predictive risk-maps that give the risk for different maneuvers. Other traffic participants are modeled with a Foresighted Driver Model. Based on this maps, an RRT* algorithm is used to find a risk and utility optimal motion plan.

Brechtel *et al.* [6] present a framework for behavioral planning by modeling the problem as a Markov Decision Process. The evolution of traffic situations is modeled with the help of dynamic bayesian networks. In [7] the authors extend this framework in order to account for partial observability with a POMDP solver that automatically learns a suited state representation based on the given problem. One major problem with their approach is the necessity for offline training of the POMDP for a specific scenario.

In [8], Ulbrich and Maurer focus on the tactical choice of making a lane change or not. They use a dynamic bayesian net as a measurement model and a tree-based policy evaluation to find the action with the best reward.

In [9], During and Pascheka give a definition of *cooperative behavior* using a utility function. They propose a recursive algorithm to generate cooperative motion plans in highway lane merging scenarios based on motion primitives, but all combinations have to be evaluated. They refine their idea in [10]: First, they initiate the solution for each vehicle independently, whether and when they want to perform a lane change depending on measures like to time to collision or time until lane ends. If independent solutions are in conflict (collision) they pairwise find a solution that does not conflict. Their goal is to minimize a global cost function. As the algorithm is applied to structured environments like highway situation a set of motion primitives represent all possible maneuvers of a vehicle.

Wei *et al.* [11] maps *socially cooperative driving* attitudes of human drivers to autonomous vehicles performing an intention prediction for surrounding vehicles and evaluating several driving strategies with a cost function based approach. A social behavior is achieved by leveraging the own and the other vehicles' goals. The presented iPCB algorithm is compared to a standard ACC and geographic information integrated ACC in terms of vehicle speeds and longitudinal and lateral distances between the vehicles during a highway merging scenario.

In their work, Frese and Beyerer [12] compares four cooperative motion planning algorithms for collision avoidance, Optimized Priorities, Tree Search, MILP, Eleastic Bands in terms of runtimes and success rates. All require inter-vehicle communication. Awal *et al.* [13] propose a cooperative lane-changing algorithm assuming dense communication between all vehicles. A significant part of their work is dedicated to traffic performance measures.

Nilsson and Sjoberg [14] propose a combined model predictive control and planning approach for deciding whether lane change or overtaking maneuvers are beneficial. Similarly, Schildbach and Borrelli [15] compute lane change trajectories via Scenario Model Predictive Control. The controller needs a traffic prediction model as input. From a given traffic scene a speed and steering profile for each surrounding vehicle is generated using an interaction-aware model up to a given horizon. The work focuses rather on the controller design than on scenario generation.

III. Problem Formulation

车辆之间存在交互博弈

In this paper, we regard the problem of planning high-level maneuvers in a highway scenario in which all acting agents can influence each other. An example scenario is depicted in Fig. 2. Formally, the problem consists of:

1) A set of N vehicles $\mathcal{P} = \{P_1, ..., P_N\}$, each having a state $\boldsymbol{x}_i \in \boldsymbol{\mathcal{X}}$ consisting of the track coordinates (s, d), the velocity v, and the current lane l
2) A scene consisting of all individual states, the road and lane geometry and obstacle information $\mathcal{S}$

按照场景进行step，每一个step包含所有的信息

3) A set of M_i possible actions for vehicle P_i $\mathcal{A}_i(\mathcal{S}) = \{a_{i,1}, ..., a_{i,M_i}\}$ for a given scene state
4) A cost function J_i for vehicle P_i given his action a_i and all other actions a_j

cost是根据自身动作以及其他车的动作联合算出来的

Although all the actions are all continuous, each vehicle is allowed to perform new actions at discrete timesteps. At these time instances all vehicles decide simultaneously. The time difference between two consecutive actions of one vehicle is denoted as Δt.

所有车都是同步决策的

Definition 3.1: Two vehicles are interacting, if an action of one vehicle can directly influences the cost function of the other vehicle. This interaction must be mutual.

Definition 3.2: Vehicle A is influenced by Vehicle B, if the action of vehicle B directly influences the cost function of vehicle A.

定义3.1:两辆车是相互作用的，如果一辆车的一个动作可以直接影响另一辆车的成本函数。这种互动必须是相互的。定义3.2:车辆B的行为直接影响到车辆A的成本函数，则车辆A受到车辆B的影响。

Thus for the planning problem, we differentiate between three different groups of vehicles depicted in Fig. 2

- the ego vehicle which we are planning for denoted as P_1 (marked red in the image).
- all vehicles directly interacting with the ego vehicle (marked blue).
- all vehicles influenced by the ego vehicle or influencing the ego vehicle (marked white).

We treat each of these groups differently in the search process in the next section.

The goal is to find a sequence of actions for the ego vehicle

$$\{a_1(t=0), a_1(t=\Delta t), \cdots, a_1(t=T_{\text{final}})\} \tag{1}$$

目标就是找到一组最优的action，使得J最小，同时假设其他的车辆是理性的，我们的最优也要尽可能节省他们的成本

that minimizes J_1, under the assumption, that all other vehicles P_i marked as interacting are acting *rational*[1], i.e. also trying to minimize their costs J_i.

IV. TCMP-MCTS Algorithm

In order to find a solution to the problem definition in the last section, we propose an algorithm TCMP-MCTS as a variant of Monte Carlo Tree Search (MCTS) that calculates tactical and cooperative motion plans. The next sections first

[1]in a game-theoretic sense

•所有与自车直接互动的车辆(蓝色标记)。

•所有受自车影响或影响自车的车辆(白色标记)。

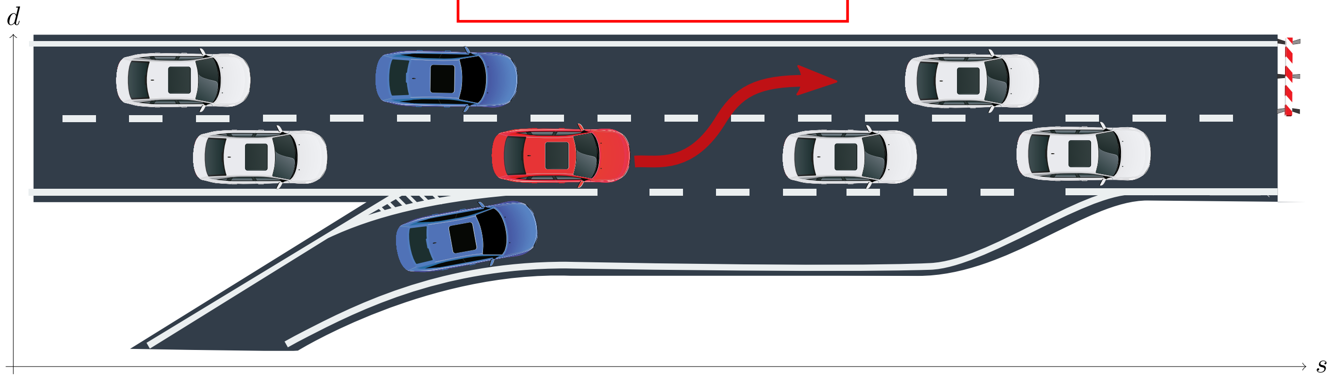

Fig. 2. Merging scene $\mathcal{S}$ describing all the relevant information for planning. The scene consists of 8 vehicles, 3 lanes (with one lane ending) and an obstacle at the end of the left lane. Red depicts the ego-vehicle. Blue are vehicles that are interacting directly with the ego vehicle.

introduce briefly the MCTS basic algorithm and then the modifications necessary for TCMP-MCTS.

A. Basic Algorithm

Monte-Carlo Tree Search (MCTS) is an algorithm to find an optimal decision with the help of random (Monte-Carlo) samples. A good overview on this topic can be found [16], which also describes the basic algorithm and several modifications made in recent literature. Here, we only outline briefly the main ideas of MCTS:

First of all, a *selection* process is done based on an existing search tree. The tree is traversed based on a *tree policy* to decide at each branch which direction to take until a leaf node is reached. Then, an *expansion* of the leaf node with one of the remaining possible actions is made to get a new leaf node. From this node, a *simulation* (often also called *rollout*) is performed with some kind of *default policy* (i.e. default behavior of all involved players) over a fixed horizon or until a terminal node is reached. Based on some *cost function* that evaluate the result of the simulation, the value of all nodes that have been traversed are updated. This whole process is depicted in Fig. 3.

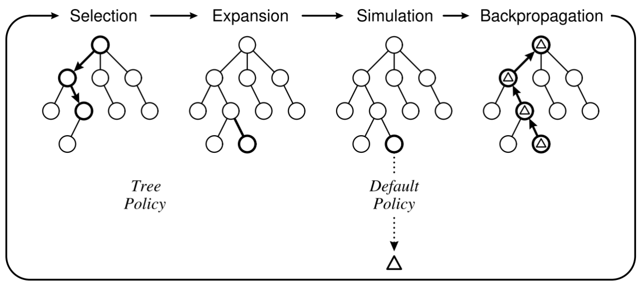

Fig. 3. Basic MCTS Search Algorithm. Image from [16]

.

Each of these steps can now be individually tuned for the problem domain at hand. The modifications that contribute to this paper are presented in the following section.

There are several advantages of MCTS, that makes it particularly useful for decision making in automated driving functions, namely:

- Anytime: MCTS can always be stopped and a result is available. It might not be optimal but it is valid.
- Parallel: MCTS can be highly parallelized, either many iterations at once, or multiple simulations in one iteration.
- Cooperative/Adversarial: MCTS can generate cooperative or adversarial behaviors depending on the cost functions.
- Versatile: MCTS can account for different planning strategies for different traffic participants (Each node and each action can have different implementations). Thus it is easily extendable.

Furthermore, the framework is easily extendible. Silver and Veness [17] for example show, how MCTS can be altered for planning in large POMDPs. Thus, for future work, also the partial observability and stochasticity of other traffic participants could be taken into account.

B. Extensions for Autonomous Driving Domain

In this section, we present the adjustments to the standard MCTS algorithm to make it applicable to the domain of cooperative planning in highway scenarios.

1) Selection: For selection we use the standard UCB1 algorithm [16]. To calculate the UCB value, we normalize all utilities such that they lie between 0 and 1 at each decision node.

2) Simultaneous Actions and Information Sets: As shown in section III, we split the planning problem into stages. At each stage, all vehicles decide simultaneously which action they will take. As this cannot be represented in a tree form, we need to make the moves sequential. Every vehicle decides similar to Soemers [18] which action to take sequentially one after the other and after the last one decided, a simulation for a fixed time Δt is performed. One main problem arising with this method is according to Cowling *et al.* [19] that later players can differentiate their strategy based on what earlier players decided. An example for three vehicles can be seen in Fig. 4. Here the decision for vehicle 2 can depend on the decision of vehicle 1, thus vehicle 2 can react and adjust its strategy accordingly. In order to enforce that the strategy for all nodes for player 2 in this stage are equal, we

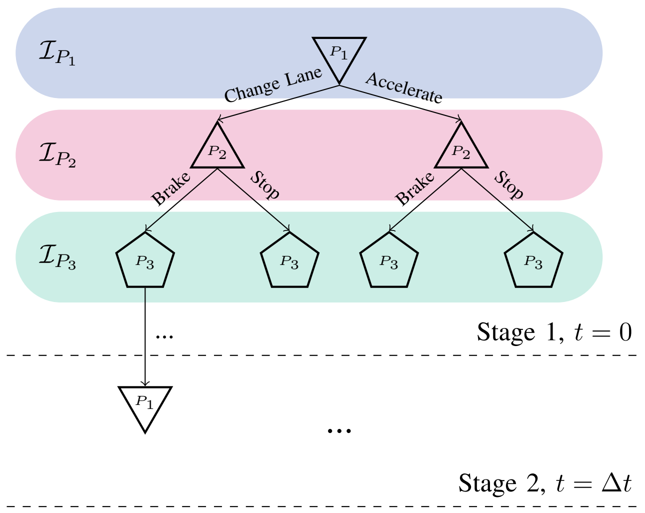

Fig. 4. Example of a decision tree for simultaneous moves with 3 vehicles. As all 3 vehicles decide at the same time instance which action to take, vehicle P_2 can not know which action P_1 took, although for constructing the tree the action was performed first. Thus vehicle 2 can be in either node, i.e. it only knows that it is in the information set $\mathcal{I}_{P_2}$. The same reasoning applies for more vehicles.

count the statistic[2] in the rollout phase for all nodes together. All nodes for one vehicle P_i sharing the same statistic are grouped together in a so-called information set $\mathcal{I}_{P_i}$. Thus this set contains all nodes, that the vehicle cannot differentiate based on the information it has. As actions in the same stage are hidden until they are revealed during simulation, this method can represent simultaneous moves. As the strategy for a node is only dependent on the statistic gathered in the backpropagation phase, all nodes in an information set share the same strategy.

默认行为是假设所有的车辆都是IDM

3) Default Policy: For the default behavior, we assume for the scope of this paper, that all vehicles behave like the Intelligent Driver Model [20]. This gives a reasonable simulation of the traffic scene over the planning horizon, if no car is further changing lanes. The equations of longitudinal motion for this model are

$$\begin{aligned} \ddot{s}_i = \quad & a\left[1 - \left(\frac{v_\alpha}{v_0}\right)^\delta\right] - a\left(\frac{s^*(v_\alpha, \triangle v_\alpha)}{s_\alpha}\right)^2, \\ s^*(v, \triangle v) = \quad & s_0 + Tv + \frac{v \triangle v}{2 * \sqrt{ab}}. \end{aligned} \tag{2}$$

with the parameters for desired velocity v_0, a time gap of T, maximal acceleration a, comfortable deceleration b, minimal distance to the front vehicle of s_0 and the acceleration exponent δ, a tuning parameter of the model.

Note, that for this step, also other models for microscopic traffic simulation could be used, or even having a stochastic model learned from real-world data.

4) Available Actions and Expansion: We define a set of possible high-level actions similar to motion primitives defined in the work of [9], [10]. The set is listed in Table I. At each node in the tree, it is first checked for each action

[2]all necessary values for the selection phase. For UCB1 this is the visitation count and mean utility.

TABLE I

POSSIBLE ACTIONS, THE PRECONDITIONS FOR ACTION TO BE AVAILABLE AND THE DESCRIPTION FOR EACH TRAFFIC PARTICIPANT

Action	*Preconditions*	*Description*
a_{i,v_0}	-	Keep the current velocity constant
$a_{i,v_0,\pm}$	-	Increase or Decrease the current velocity with a fixed acceleration a_{acc}
$a_{i,T}$	Leading vehicle	Keep time gap constant to leading vehicle
$a_{i,Ⓢ}$	Stopping point	Come to a stop at a stopping location. This could be the end of lane, an intersection entering, an obstacle ahead, ...
$a_{i,\rightleftarrows}$	Lane left/right	Sets reference lane parameter to the left or right lane to perform a lane change

定义一些高级的动作集用于推演，每次选择时先判断该场景下能不能选

if the preconditions are met. During integration, different dynamical models are chosen for the longitudinal motion depending on the action: For a_{i,v_0} and $a_{i,v_0,\pm}$ a simple point mass dynamic with $\ddot{s}_i = \{0, a_{\mathrm{acc}}, -a_{\mathrm{acc}}\}$ is used. For gap keeping $a_{i,T}$ the IDM model in (2) is used. If approaching an stopping point with $a_{i,Ⓢ}$, it is set as a leader with zero velocity. For the lateral motion, we use a simple P-controller to track the lateral coordinate

横向上用比例控制器

$$\dot{d}_i = K_{P,d} \cdot (d_i - d_{\mathrm{ref}}) \tag{3}$$

where $K_{P,d}$ is a design parameter and controls how fast a lane change should be and d_{ref} denotes the lateral offset of the currently tracked lane.

5) Cost Function and Backpropagation: For each of the relevant vehicles P_i, we define a cost function J_i consisting of the terms

每一辆车都会有个成本。

- $J_{i,LC}$: Costs for performing a lane change
- $J_{i,v}$: Costs for differing from desired velocity $\sum \|v_i - v_0\|$
- $J_{i,a}$: Costs for accelerations $\sum a_i^2$
- $J_{i,o}$: Costs for distance to obstacles and other vehicles in longitudinal direction $\sum \frac{1}{d_{i,obst}}$
- $J_{i,\mathrm{inv}}$: Very high costs for reaching an invalid state, for example a collision.

All of these cost terms are summed up as

每个成本都有各自的权重

$$J_i = \sum w_{i,(\cdot)} J_{i,(\cdot)} \tag{4}$$

with weight parameters $w_{i,(\cdot)}$ for each cost term. We further introduce a cooperative cost function for vehicle i as

构建一个联合的成本函数，考虑他车

$$J_{i,\mathrm{coop}} = J_i + \sum_{j \neq i} \lambda \cdot J_j \tag{5}$$

with a cooperation factor λ. This parameter allows an adjustment of each traffic participant how cooperative it will be. $\lambda = 0$ means complete egoistic behavior and $\lambda = 1$ will weigh other vehicles' costs equally and thus will produce a solution that optimizes the global utility. Values in between can show different levels of cooperation. This cooperative cost function is used to update the MCTS search tree during backpropagation.

有一个配合因子λ。此参数允许调整每个流量参与者的合作程度。λ = 0表示完全的利己行为，λ = 1将平均权衡其他车辆的成本，从而产生一个优化全局效用的解决方案。两者之间的值可以显示不同程度的合作。该合作代价函数用于在反向传播过程中更新MCTS搜索树。

In contrast to our work, Schwarting and Pascheka [10] also add costs for driving on an ending lane and costs for

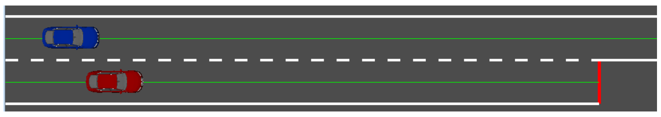

Fig. 5. Two lane merging scene with two vehicles having equal speed

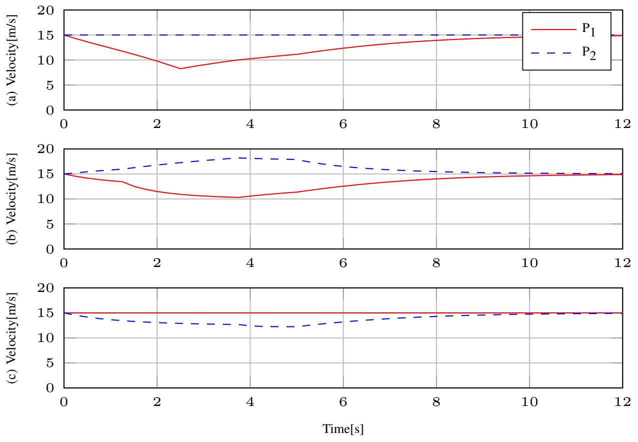

Fig. 6. Velocity Profile for both vehicles for the three cases (a),(b),(c) with $\lambda = 0$, $\lambda = 0.5$ and $\lambda = 1$ respectively.

overtaking on the right side. In [9] on the other hand, only longitudinal and lateral accelerations of the vehicles are used to calculate costs.

6) Considered Vehicles and Decision Nodes: As mentioned in the previous section, we have different sets of vehicles in the planning process. First of all, only vehicles within a distance to the ego-vehicle are considered in the planning process at all. For all vehicles that are interacting (red and blue in Fig. 2), we use decision node with actions as presented in section IV-B.4. For all the white vehicles, we always take the default policy, i.e. integrating their behavior with the IDM-model. But for all vehicles, the cost function is calculated and affects the cooperative cost function for the ego vehicle.

对不同的车考虑应该也是不一样的

交互性强的车采用action集合来推演

交互性不强的车采用默认策略进行推演

7) Terminal Nodes: A node is marked as terminal, if a time horizon T_{final} is reached, or if the state is invalid, for example if a collision occurred.

7)终端节点:如果到达时间范围Tfinal，或者如果状态无效，例如发生碰撞，则节点被标记为终端。

V. Experiments

In order to examine the behavior of our proposed algorithm in real world scenarios, we perform several simulations. We consider a two or three lane road with the right most lane ending. The ego-vehicle drives on this ending lane. An example is depicted in Fig. 5.

A. Merging to the left in a two vehicle scenario on a two lane road

In this simple setup the right lane is ending and the vehicle P_1 has to merge to the left lane. We examine three possible solutions for the conflict situation;

(a) vehicle P_2 keeps its velocity and vehicle P_1 has to decelerate. This behavior can reasonably called non-cooperative.
(b) vehicle P_2 accelerates smoothly giving vehicle P_1 the possibility to merge in behind of P_2 with lower speed.
(c) vehicle P_2 decelerates smoothly giving vehicle P_1 the possibility to keep its velocity and merge in front of P_2.

Planning and simulating the scene with changing cooperation parameter λ, we obtain the three solutions. The velocity profiles for both vehicles can be found in Fig. 6. In the plots we see, that for scenario (a) the ego vehicle must brake strongly in order to avoid reaching the end of the lane. On the other hand, vehicle P_2 is simply keeping the velocity at the desired velocity. For scenario (b) both vehicles must change their speed slightly in order to avoid each other. In (c), it is sufficient, that P_2 is braking a little bit in order to let P_1 merge. For the situations we obtain following cost values:

Scenario	λ	J_1	J_2	J_{total}
(a)	0	141.7	0	141.7
(b)	0.5	90.6	25.7	116.3
(c)	1	10.1	62.2	73.3

We see that with an increasing λ the total costs decrease, but the cost for P_2 increases. In [9], cooperative behavior is categorized as egoistic, altruistic, or rational. Taking scenario (a) as a baseline where no cooperation is taking place, (b) and (c) show altruistic behavior of vehicle P_2. This means it is decreasing the total costs by increasing its owns costs. Because of these findings, we choose $\lambda = 1$ for the next sections to further analyze the cooperative behavior of this algorithm.

B. Merging to the left in a four vehicle scenario on a two lane road

Similar to section V-A the right lane is ending and the blue vehicle has to merge on the left lane into a steady traffic flow. In Fig. 7, we can see a series of time-snapshots how the scenario evolves during a simulation. In contrast to the previous scenario, the ego vehicle needs to make space for a merger. As we can see in the images, the ego vehicle indeed increases the gap to its leader to let the blue vehicle merge. The white vehicles behind the ego vehicle need to brake also. After some time, the gaps between the vehicles gradually converge to be equally spaced again. For this scenario, two things can be noted: First of all, any motion planning algorithm that does not consider the mutual influence would plan a stop at the end of the lane as the gap for merging is too small. Second, with TCMP-MCTS, the disturbance of the other vehicles is minimized and the overall traffic flow is optimized.

首先，任何不考虑相互影响的运动规划算法都会因为合并间隙太小而在车道的末端进行停车规划。其次，采用tcp - mcts可以最大限度地减少其他车辆的干扰，优化整体交通流。

C. Merging and letting merge on a three lane road with multiple vehicles with congested traffic

Next, we consider a scenario of merging onto a street as depicted in Fig. 8. Here, a situation with 3 interacting vehicles is considered. The ego vehicle anticipates that the blue vehicle on the middle lane might change the lane to make space. Thus, the ego vehicle keeps its speed instead

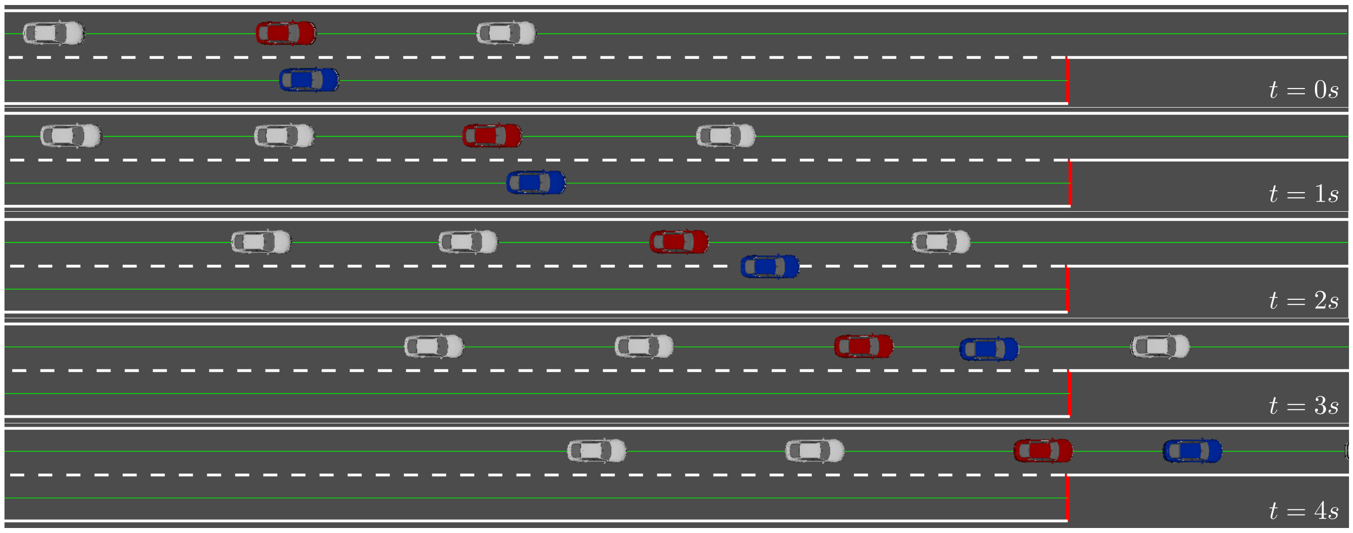

Fig. 7. Two lane merging scene with multiple vehicles. The ego vehicle is reducing its speed in order to let the blue vehicle merge in front of it.

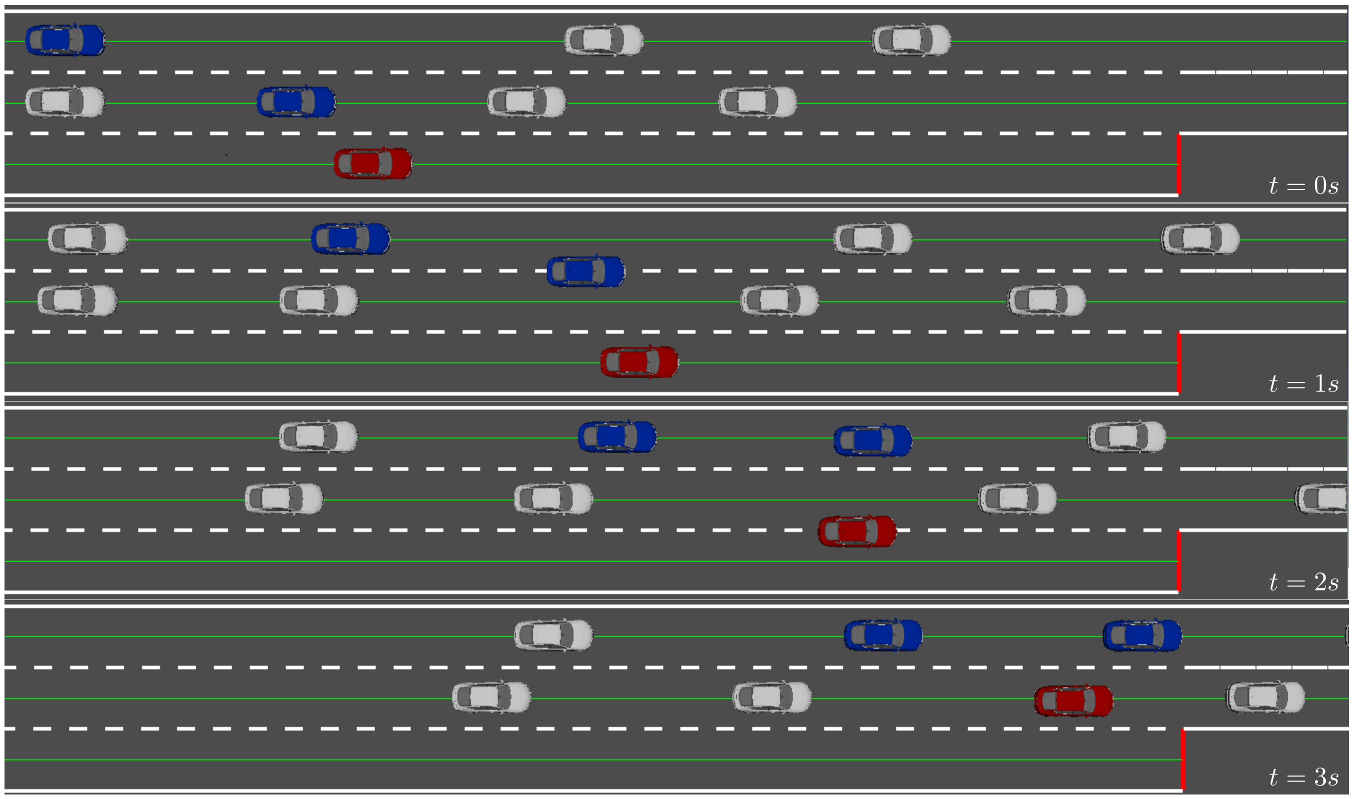

Fig. 8. Merging scene with congestion with multiple vehicles. A lane change is necessary to make space for the red ego vehicle.

of braking for the first second. After sensing, that the blue vehicle has in fact changed the lane, the ego vehicle now can safely merge without too much disturbance of the traffic flow.

D. Discussion

In the previous simulations, we assumed that all traffic participants behave rational and perform the action with the lowest cost for them. In reality, the cost function of other vehicles is not clear and other vehicles make suboptimal decisions. However, by applying MCTS with UCB1 selection, we found during our experiments that the plan is robust against such variations. The reason for this is twofold. First, if two actions have (estimated) equal costs, then they are explored about the same number of times. Thus, the prior node in the search tree has to assume that the next action is chosen randomly between those two actions and will therefore accumulate the mean utility. The second reason for the robustness lies in the nature how a plan is found: The result in form of a MCTS tree is a feedback plan indicating what the ego vehicle should do depending on the action of others. Thus, even if a vehicle chooses a suboptimal action, the MCTS-tree has the best response to this action included. Although this is not a proof, this gives a strong indication towards robustness. We plan to examine this point further with the help of real-world driving data in future work.

VI. Conclusion and Future Work

In this paper, we presented the algorithm TCMP-MCTS that considers interactions between different vehicles in order to plan cooperative motion plans. We have shown in a case-study, that with this algorithm it is possible to generate cooperative behaviors that can be observed at human driving styles: Speeding up, braking, or making a lane change to let somebody merge. This behavior was not hardcoded into the system, but emerged from considering possible actions of others and their cost functions.

For future work, several directions are imaginable: The trajectories calculated from the presented approach are not optimal in terms of comfort. As Schwarting and Pascheka [10] propose, a jerk minimizing trajectory generation concept can be used to compute smooth trajectories. Furthermore, we need to make further quantitative evaluations regarding the cost function and add possibly cost on macroscopic measures as the vehicles contribution to the traffic flow. Additionally comparisons with other approaches in reference scenarios have to be implemented.

References

[1] S. Ulbrich, S. Grossjohann, C. Appelt, *et al.*, "Structuring Cooperative Behavior Planning Implementations for Automated Driving," in *IEEE Intelligent Transport Systems Conference*, 2015.

[2] M. Montemerlo, J. Becker, S. Bhat, *et al.*, "Junior: The stanford entry in the urban challenge," *Journal of Field Robotics*, vol. 25, no. 9, pp. 569–597, 2008.

[3] S. Kammel, J. Ziegler, B. Pitzer, *et al.*, "Team AnnieWAY's autonomous system for the 2007 DARPA Urban Challenge," *Journal of Field Robotics*, vol. 25, no. 9, pp. 615–639, 2008.

[4] A. Bacha, C. Bauman, R. Faruque, *et al.*, "Odin: Team VictorTango's entry in the DARPA Urban Challenge," en, *Journal of Field Robotics*, vol. 25, no. 8, pp. 467–492, 2008.

[5] F. Damerow and J. Eggert, "Risk-aversive behavior planning under multiple situations with uncertainty," in *IEEE Intelligent Transport Systems Conference*, 2015.

[6] S. Brechtel, T. Gindele, and R. Dillmann, "Probabilistic MDP-behavior planning for cars," in *IEEE Intelligent Transport Systems Conference*, 2011, pp. 1537–1542.

[7] ——, "Probabilistic Decision-Making under Uncertainty for Autonomous Driving using Continuous POMDPs," in *IEEE Intelligent Transport Systems Conference*, 2014.

[8] S. Ulbrich and M. Maurer, "Towards Tactical Lane Change Behavior Planning for Automated Vehicles," in *IEEE Intelligent Transport Systems Conference*, 2015.

[9] M. During and P. Pascheka, "Cooperative Decentralized Decision Making for Conflict Resolution among Autonomous Agents," in *IEEE International Symposium on Innovations in Intelligent Systems and Applications (INISTA)*, 2014.

[10] W. Schwarting and P. Pascheka, "Recursive Conflict Resolution for Cooperative Motion Planning in Dynamic Highway Traffic," in *IEEE Intelligent Transport Systems Conference*, 2014.

[11] J. Wei, J. M. Dolan, and B. Litkouhi, "Autonomous Vehicle Social Behavior for Highway Entrance Ramp Management," *IEEE Intelligent Vehicles Symposium*, 2013.

[12] 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.

[13] T. Awal, M. Murshed, and M. Ali, "An efficient cooperative lane-changing algorithm for sensor- and communication-enabled automated vehicles," in *IEEE Intelligent Vehicles Symposium*, 2015.

[14] J. Nilsson and J. Sjoberg, "Strategic decision making for automated driving on two-lane, one way roads using model predictive control," in *IEEE Intelligent Vehicles Symposium*, 2013, pp. 1253–1258.

[15] G. Schildbach and F. Borrelli, "Scenario Model Predictive Control for Lane Change Assistance on Highways," in *IEEE Intelligent Vehicles Symposium*, 2015.

[16] C. Browne, E. Powley, D. Whitehouse, *et al.*, "A survey of monte carlo tree search methods," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 4, no. 1, 2012.

[17] D. Silver and J. Veness, "Monte-Carlo Planning in Large POMDPs," in *Advances in Neural Information Processing Systems*, 2010, pp. 2164–2172.

[18] D. Soemers, "Tactical Planning Using MCTS in the Game of StarCraft 1 StarCraft : Brood War," Master Thesis, 2014.

[19] P. Cowling, E. Powley, and D. Whitehouse, "Information Set Monte Carlo Tree Search," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 4, no. 2, 2012.

[20] M. Treiber, A. Hennecke, and D. Helbing, "Congested Traffic States in Empirical Observations and Microscopic Simulations," *Physical Review E*, vol. 62, no. 2, 2000.