

Parallel Optimization with Hard Safety Constraints for Cooperative Planning of Connected Autonomous Vehicles

Zhenmin Huang, Haichao Liu, Shaojie Shen, and Jun Ma

Abstract—The development of connected autonomous vehicles (CAVs) facilitates the enhancement of traffic efficiency in complicated scenarios. Difficulties remain unsolved in developing an effective and efficient coordination strategy for CAVs. In this paper, we formulate the cooperative autonomous driving task of CAVs as an optimal control problem with safety conditions enforced as hard constraints, and propose a computationally-efficient parallel optimization framework to generate strategies for CAVs with the travel efficiency improved and the hard safety constraints satisfied. Specifically, all constraints involved are addressed appropriately with convex approximation, such that the convexity property of the reformulated optimization problem is exhibited. Then, a parallel optimization algorithm is presented to solve the reformulated optimization problem, with an embodied iterative nearest neighbor search strategy to determine the optimal passing sequence. It is noteworthy that the travel efficiency is enhanced and the computation burden is considerably alleviated with the proposed innovation development. We also examine the proposed method in CARLA simulator and perform thorough comparisons to demonstrate the effectiveness and efficiency of the proposed approach.

I. INTRODUCTION

Recent developments in artificial intelligence and information technologies prompt the emergence of connected autonomous vehicles (CAVs), which possess the ability to exchange information among vehicles through real-time communication technology such as 5G network and Wi-Fi [1]. In this sense, cooperative autonomous driving is enabled, which points to the alleviation of traffic congestions induced by non-cooperative driving manners, and also the decrease of traffic accidents [2]. Although the physical basis is well-prepared, a series of problems remain largely

This work was supported in part by the National Natural Science Foundation of China under Grant 62303390; and in part by the Project of Hetao Shenzhen-Hong Kong Science and Technology Innovation Cooperation Zone under Grant HZQB-KCZYB-2020083. (*Corresponding Author: Jun Ma.*)

Zhenmin Huang is with the Robotics and Autonomous Systems Thrust, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China, and also with the Division of Emerging Interdisciplinary Areas, The Hong Kong University of Science and Technology, Hong Kong SAR, China (email: zhuangdf@connect.ust.hk).

Haichao Liu is with the Robotics and Autonomous Systems Thrust, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China (e-mail: hliu369@connect.hkust-gz.edu.cn).

Shaojie Shen is with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong SAR, China (email: eeshaojie@ust.hk).

Jun Ma is with the Robotics and Autonomous Systems Thrust, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China, also with the Division of Emerging Interdisciplinary Areas, The Hong Kong University of Science and Technology, Hong Kong SAR, China, and also with the HKUST Shenzhen-Hong Kong Collaborative Innovation Research Institute, Futian, Shenzhen, China (e-mail: jun.ma@ust.hk).

unsolved in developing a mature coordination scheme that is suitable for cooperative autonomous driving in complicated urban traffic scenarios. Rule-based, optimization-based, and learning-based methods are intensively researched to produce various coordination strategies for different urban traffic scenarios, such as on-ramps merging [3], [4], [5], [6] and intersection crossing [7], [8]. However, existing methods either merely focus on simple scenarios or only consider simple interactions between a limited number of CAVs, and therefore are not readily applicable to complicated urban traffic scenarios with dense traffic flow. Besides, a common problem in existing methods is the low computation efficiency. As the number of CAVs involved in the scenario increases, the scale of the problem continues to grow, which leads to a rapid surge in the overall computation time that violates the real-time requirements for planning and control of CAVs. As a result, current methods generally fail to apply to scenarios containing a large number of CAVs.

In this paper, we focus on the development of an optimization framework to solve the cooperative autonomous driving problem for all-directional traffic flows in urban traffic scenarios. To provide hard safety guarantees, a linearization method is introduced to properly handle the complicated nonlinear collision avoidance constraints and road boundary constraints. To improve travel efficiency, a novel coordination strategy based on the iterative nearest neighbor search is proposed to determine the optimal passing sequence. In conjunction with a parallel optimization method based on dual consensus ADMM, the computational burden is greatly alleviated and therefore real-time performance is achieved. In terms of experimental validation, we compare our method with several baselines to demonstrate both the improved travel efficiency and the superiority in computational efficiency.

II. RELATED WORKS

Various methods are intensively investigated in developing the strategy for cooperative driving in urban traffic scenarios. Owing to the rapid development of data science and artificial intelligence, learning-based methods are currently progressing by leaps and bounds, which leverage real-world data to boost performance. It is declared that learning-based methods are effective to address difficult traffic scenarios [9]. In [10], a centralized coordination scheme using reinforcement learning is proposed, which allows for the cooperative crossing of CAVs at an unsignalized intersection. In [11], multi-agent reinforcement learning is adopted to enable sympathy between CAVs and human drivers for efficient

highway merging. Though with great potential, learning-based methods generally require a large amount of real-world data and they could suffer from poor interpretability, which hinders their wide applications in real-world scenarios.

In contrast to learning-based methods, rule-based and optimization-based methods are more mature and are capable of generating predictable and explainable results with limited computational resources. Particularly, rule-based methods mainly focus on the determination of passing orders of CAVs using handcrafted rules and criteria, followed by a low-level planner to generate trajectories for all CAVs. In [2], a communication-enabled distributed mechanism is proposed to resolve conflicts for CAVs in intersections. Each vehicle solves a conflict graph locally to compute the desired time slot for passing, followed by a motion planner to determine the speed profile. In [12], a centralized convex optimal control framework is proposed with a hierarchical coordination scheme to enable unsignalized intersection crossing for CAVs. In [13], a method based on Monte Carlo tree search (MCTS) and heuristic rules is proposed to find nearly global optimal orders for CAVs to pass unsignalized intersections. Although rule-based methods are straightforward, it is generally hard to cover various corner cases with man-made rules, resulting in difficulties for rule-based methods to generalize to complicated scenarios.

Meanwhile, optimization-based methods typically formulate the cooperative driving as a constrained optimization problem and adopt a unified optimization framework to plan directly for all CAVs, thus being more concise and generalizable [14], [15]. In [16], an approach based on mixed-integer quadratic programming (MIQP) is presented to obtain globally optimal solutions for cooperative driving of CAVs in general on-road scenarios. In [7], the cooperative trajectory planning problem is formulated as a mixed integer nonlinear program (MINLP) to determine conflict-free trajectories. However, these methods are only applicable to a small number of vehicles, as the computation time soon becomes unaffordable when the number of vehicles becomes large. To improve computation efficiency, distributed and parallel optimization methods are strongly desired. The alternating direction method of multipliers (ADMM) [17] provides a general distributed framework to accelerate the optimization process, which triggers its recent applications on cooperative autonomous driving [18]. In [19], [20], ADMM is used to split independent vehicle constraints from coupled collision-avoidance constraints between vehicles such that they can be dealt with separately, yielding a partially parallel optimization scheme. In [21], a fully decentralized and parallel optimization framework is proposed to solve the cooperative trajectory planning problem for general types of robots. However, its computational efficiency is still far from being satisfying, as minutes are taken for solving the problem involving only several robots. Based on dual consensus ADMM [22], a parallel optimization framework is proposed in [23], which enables cooperative trajectory planning of CAVs with high computation efficiency. However, it adopts soft penalties for collision avoidance, which results in possi-

ble collisions or over-conservative strategies if penalties are not set appropriately. Besides, the devised approach neglects certain pertinent constraints for driving safety such as road boundary limitations. These problems render the generated strategy potentially unsafe in real-world situations. Also, the time profiles for all CAVs are preassigned non-cooperatively and fixed, which may cause difficulties in resolving conflicts for situations like lane merging.

III. PROBLEM STATEMENT

A. Vehicle Kinematics and Physical Constraints

For a traffic scenario involving N CAVs passing simultaneously, we use $\mathcal{N} = \{1, 2, \dots, N\}$ to denote the set containing indices of all CAVs. We assume a complete graph structure on the communication network, and all CAVs involved possess the same kinematics, which can be expressed as

$$x_{\tau+1}^i = f(x_{\tau}^i, u_{\tau}^i), \quad (1)$$

where x_{τ}^i and u_{τ}^i are the state vector and the control input vector corresponding to vehicle i at timestamp τ , $\tau \in \mathcal{T} = \{0, 1, \dots, T-1\}$, and T is the length of the horizon. In particular, we consider a state vector of four state variables $x_{\tau}^i = (p_{x,\tau}^i, p_{y,\tau}^i, \theta_{\tau}^i, v_{\tau}^i)$ and an input vector constituted by two control inputs $u_{\tau}^i = (\delta_{\tau}^i, a_{\tau}^i)$, where $p_{x,\tau}^i$ and $p_{y,\tau}^i$ denote the X and Y coordinates of the midpoint of the rear axle in the global frame, θ_{τ}^i denotes the heading angle, v_{τ}^i denotes the velocity, δ_{τ}^i represents the steering angle, and a_{τ}^i represents the acceleration. With the above notations, we adopt the following kinematic model [24] for vehicle i :

$$\begin{cases} p_{x,\tau+1}^i = p_{x,\tau}^i + f_r(v_{\tau}^i, \delta_{\tau}^i) \cos(\theta_{\tau}^i), \\ p_{y,\tau+1}^i = p_{y,\tau}^i + f_r(v_{\tau}^i, \delta_{\tau}^i) \sin(\theta_{\tau}^i), \\ \theta_{\tau+1}^i = \theta_{\tau}^i + \arcsin(\tau_s v_{\tau}^i \sin(\delta_{\tau}^i)/b), \\ v_{\tau+1}^i = v_{\tau}^i + \tau_s a_{\tau}^i. \end{cases} \quad (2)$$

Note that b is the vehicle wheelbase and τ_s is the time interval. The function $f_r(v, \delta)$ is defined as

$$f_r(v, \delta) = b + \tau_s v \cos(\delta) - \sqrt{b^2 - (\tau_s v \sin(\delta))^2}. \quad (3)$$

Meanwhile, the following constraints are imposed on the control inputs:

$$a_{\min} \leq a_{\tau}^i \leq a_{\max}, \quad \delta_{\min} \leq \delta_{\tau}^i \leq \delta_{\max}. \quad (4)$$

B. Collision Avoidance Constraints

To ensure driving safety, constraints on minimal distances between CAVs must be imposed to avoid any possible collision. We make the assumption that all CAVs are of the same size, and approximate each vehicle with two identical circles aligned in the longitudinal direction. The front circle and the rear circle of vehicle i are denoted as FC_i and RC_i , respectively. With such an approximation, a sufficient condition for collision avoidance is that the distance between the centers of any pair of circles on different vehicles is no less than the diameter. We denote the center positions of

FC_i and RC_i at time τ as $p_{f,\tau}^i$ and $p_{r,\tau}^i$, respectively. The collision-free condition is thus expressed as

$$\|p_{\beta,\tau}^i - p_{\gamma,\tau}^j\|_2 \geq d_{\text{safe}}. \quad (5)$$

Notice that d_{safe} is the safe distance to maintain, which is set to be equal to the common diameter of all circles.

C. Road Boundary Constraints

To ensure that all CAVs are driving within the free space, road boundary constraints must be imposed. A sufficient condition for the satisfaction of road boundary constraints is that the minimal distance from the centers of circles to the corresponding closest road boundaries should always be no less than the radius of all circles. Equivalently, we shift all road boundaries towards the internal of the free space by a bias that equals the common radius of all circles. As a result, a new set \mathcal{C} is formed, which is a subset of the free space. Then the following constraints are established:

$$p_{\beta,\tau}^i \in \mathcal{C}. \quad (6)$$

D. Overall Formulation

For CAVs, we set the goal as the minimization of the deviation of actual trajectories from reference trajectories, as well as the control inputs. Integrating all the constraints mentioned above, we obtain the optimal control problem as below:

$$\begin{aligned} \min_{\{u_\tau^i\}} & \sum_{i=1}^N \left(\sum_{\tau=0}^T \|x_\tau^i - x_{\tau,ref}^i\|_{Q_i}^2 + \sum_{\tau=0}^{T-1} \|u_\tau^i\|_R^2 \right) \\ \text{s.t.} & (1), (4), (5), (6), \\ & \forall \beta, \gamma \in \{f, r\}, i, j \in \mathcal{N}, i \neq j, \tau \in \mathcal{T}. \end{aligned} \quad (7)$$

$x_{\tau,ref}^i$ is the reference point for vehicle i at timestamp τ , which contains reference waypoint, reference heading angle, and reference velocity. $\|\cdot\|_Q$ and $\|\cdot\|_R$ denote the L_2 -norm weighted by matrices Q and R , where Q is positive semi-definite and R is positive definite.

IV. METHODOLOGY

In this section, we propose an optimization framework for solving problem (7) to obtain the optimal cooperative driving strategy. With a slight abuse of notations, we define the current nominal trajectories as $\{x_\tau^i, u_\tau^i\}$, for $\tau \in \mathcal{T}$ and $i \in \mathcal{N}$. Our goal is to search for the optimal variations $\{\delta x_\tau^i, \delta u_\tau^i\}$ on the nominal trajectories that minimize the objectives while satisfying the constraints. This process needs to be iterated until trajectories for all CAVs finally converge.

A. Linearization of Hard Safety Constraints

Safety constraints involved in urban traffic scenarios include collision avoidance constraints and road boundary constraints. It should be noted that both are imposed on the center positions of FC and RC , while the optimization variables are bound to the rear axle centers of CAVs. This gap needs to be bridged by the Jacobians in-between, namely

$$\delta p_{\beta,\tau}^i = J_{\beta,\tau}^i \delta x_\tau^i. \quad (8)$$

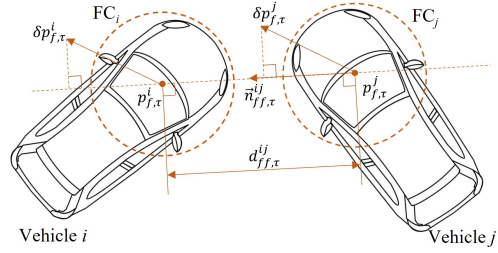


Fig. 1. Linearization of collision avoidance constraints.

The original collision avoidance constraints are obviously non-convex and nonlinear, which could lead to difficulties in finding a feasible solution. Similar to [18], a straightforward method to render these constraints linear is adopted. For each pair of circles between vehicle i and vehicle j , we construct a unit vector $\bar{n}_{\beta\gamma,\tau}^{ij}$ pointing from the center of one circle to the other. Then the following inequality provides a sufficient condition for collision avoidance between the two circles:

$$\bar{n}_{\beta\gamma,\tau}^{ij,\top} (J_{\beta,\tau}^i \delta x_\tau^i - J_{\gamma,\tau}^j \delta x_\tau^j) + d_{\beta\gamma,\tau}^{ij} - d_{\text{safe}} \geq 0. \quad (9)$$

The principle behind this inequality is that the difference between the projections of $\delta p_{\beta,\tau}^i$ and $\delta p_{\gamma,\tau}^j$ onto $\bar{n}_{\beta\gamma,\tau}^{ij}$ gives the lower bound of change in current distance $d_{\beta\gamma,\tau}^{ij}$, as is shown in Fig. 1. Therefore, the collision-free condition is satisfied when the sum of this difference and the current distance is greater than or equal to d_{safe} .

To handle irregular road boundaries, we propose a linearization method that can theoretically be applied to road boundary constraints of any shape. We first extract all road boundaries in the scenario and then perform dense sampling on the boundaries to turn them into a point cloud, denoted as \mathcal{M} . Then, the nearest neighbor search is performed on \mathcal{M} to obtain the closest boundary point corresponding to the vehicle location at each timestamp:

$$p_{\beta b,\tau}^i \leftarrow \text{NearestNeighbor}(\mathcal{M}, p_{\beta,\tau}^i), \quad (10)$$

where $p_{\beta b,\tau}^i$ is the point closest to $p_{\beta,\tau}^i$ within \mathcal{M} . We then construct the unit vector pointing from $p_{\beta b,\tau}^i$ to $p_{\beta,\tau}^i$, denoted as $\bar{n}_{\beta,\tau}^i$. With the same logic of (9), the linearized road boundary constraints can be expressed as

$$2\bar{n}_{\beta,\tau}^{i,\top} J_{\beta,\tau}^i \delta x_\tau^i + 2d_{\beta,\tau}^i - d_{\text{safe}} \geq 0. \quad (11)$$

Fig. 2 demonstrates the linearized road boundary constraints. To obtain the closest point efficiently, we perform the nearest neighbor search using the k-d tree, which exhibits $O(\log N)$ complexity for a single query to a point cloud containing N points.

B. Implicit Passing Sequence Coordination

It is a common situation in urban traffic scenarios that two vehicles try to merge into one lane simultaneously. This gives rise to the critical problem of determining the passing sequence. In this part, we propose a novel strategy that obtains the optimal passing sequence implicitly through iterative constructions of objective functions. We first define

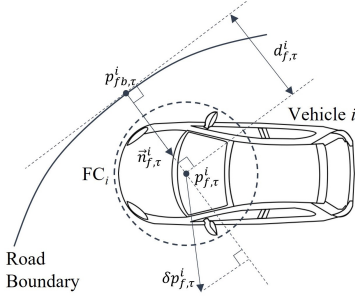


Fig. 2. Linearization of road boundary constraints.

$\delta X^i = (\delta x_0^i, \delta u_0^i, \dots, \delta x_T^i)$ as the concatenated vector of all variational states and inputs corresponding to vehicle i . From (7), the objective function corresponding to vehicle i is

$$C^i(\delta X^i) = \sum_{\tau=0}^T \|\delta x_\tau^i\|_{Q_\tau}^2 + 2(x_\tau^i - x_{\tau,ref}^i)^\top Q_\tau \delta x_\tau^i + \sum_{\tau=0}^{T-1} \|\delta u_\tau^i\|_R^2 + 2u_\tau^{i\top} R \delta u_\tau^i. \quad (12)$$

Since the passing order of vehicle i is not determined, the reference point $x_{\tau,ref}^i$ corresponding to timestamp τ is unknown. To obtain the optimal reference waypoint, we adopt the nearest neighbor search. Suppose the reference path generated by the navigation module for vehicle i is $\{p_{ref}^i\}$, which is the set of all waypoints contained in the path. The current reference waypoint for time τ is given as

$$p_{\tau,ref}^i \leftarrow \text{NearestNeighbor}(\{p_{ref}^i\}, p_\tau^i), \quad (13)$$

where $p_\tau^i = (p_{x,\tau}^i, p_{y,\tau}^i)$. Nearest neighbor search and reconstruction of the objective functions are performed at each iteration of optimization. As a result, the vehicle that already takes the lead will set its own reference waypoints further ahead, while the vehicle that falls behind does the opposite. Together with the safety constraints, they are iteratively separated; and as a result, the passing sequence is determined.

The nearest neighbor search for the closest waypoint is also implemented using the k-d tree for improved efficiency. Constant reference velocity v_{ref} is used for all vehicles at all time stamps, and no restriction on heading angles is imposed. Moreover, covariance matrix Q_τ^i is defined such that only lateral deviation from the reference path is considered, while longitudinal deviation from the reference waypoint along the reference path is ignored.

C. Optimization Problem Reformulation

To handle the nonlinear kinematics, we perform the first-order Taylor expansion of (1) to obtain

$$\delta x_{\tau+1}^i = A_\tau^i \delta x_\tau^i + B_\tau^i \delta u_\tau^i. \quad (14)$$

Input constraints (15) are reformulated as

$$a_{\min} - a_\tau^i \leq \delta a_\tau^i \leq a_{\max} - a_\tau^i, \delta_{\min} - \delta_\tau^i \leq \delta \delta_\tau^i \leq \delta_{\max} - \delta_\tau^i. \quad (15)$$

Algorithm 1 Dual Consensus ADMM [22]

- 1: **choose** $\sigma, \rho > 0$
- 2: **initialize** for all $i \in \mathcal{N}$: $p^{i,0} = y^{i,0} = z^{i,0} = s^{i,0} = 0$
- 3: **repeat**: for all $i \in \mathcal{N}$
- 4: Broadcast $y^{i,k}$ to all other vehicles
- 5: $p^{i,k+1} \leftarrow p^{i,k} + \rho \sum_{j \neq i} (y^{i,k} - y^{j,k})$
- 6: $s^{i,k+1} \leftarrow s^{i,k} + \sigma (y^{i,k} - z^{i,k})$
- 7: $r^{i,k+1} \leftarrow \rho \sum_{j \neq i} (y^{i,k} + y^{j,k}) + \sigma z^{i,k} - p^{i,k+1} - s^{i,k+1}$
- 8: $x^{i,k+1} \leftarrow \arg \min \{f^i(x^i) + \eta \|J^i x^i + r^{i,k+1}\|^2\}$
- 9: $y^{i,k+1} \leftarrow 2\eta (J^i x^{i,k+1} + r^{i,k+1})$
- 10: $z^{i,k+1,*} \leftarrow \text{Prox}_{\frac{1}{\sigma}}^{N^i} (N(s^{i,k+1} + \sigma y^{i,k+1}))$
- 11: $z^{i,k+1} \leftarrow \frac{s^{i,k+1}}{\sigma} + y^{i,k+1} - \frac{1}{N^i} z^{i,k+1,*}$
- 12: **until** termination criterion is satisfied

We tile up (9), (11), and (15) and rewrite them into a compact matrix form, which yields

$$\sum_{i=1}^N J^i \delta X^i + l \geq 0. \quad (16)$$

We denote the indicator function over a given set \mathcal{S} as $\mathcal{I}_{\mathcal{S}}(x)$. With this notation, we gather (16), objective functions, and linearized kinematics to obtain the following quadratic optimization problem

$$\min_{\{\delta X^i\}} \sum_{i=1}^N (C^i(\delta X^i) + \mathcal{I}_{\mathcal{K}^i}(\delta X^i)) + \mathcal{I}_{R_+^n} \left(\sum_{i=1}^N J^i \delta X^i + l \right), \quad (17)$$

where \mathcal{K}^i is the feasible set for linearized kinematics of vehicle i and R_+^n is the nonnegative orthant with n being the length of l .

D. Parallel Optimization

Dual consensus ADMM is adopted to perform parallel optimization. We define the proximal operator as $\text{Prox}_f^{\rho}(x) = \arg \min_y \{f(y) + \frac{\rho}{2} \|x - y\|^2\}$, and Algorithm 1 [22] can be used to solve the following convex optimization problem:

$$\min_{\{x^i\}} \sum_{i=1}^N f^i(x^i) + g \left(\sum_{i=1}^N J^i x^i \right). \quad (18)$$

Note that p, s, r, y, z are column vectors of the same size as l , and $\eta = 1/(2(\sigma + 2\rho(N-1)))$. We apply dual consensus ADMM to problem (17) with the following substitution: $x^i \leftarrow \delta X^i$, $f^i(\cdot) \leftarrow C^i(\cdot) + \mathcal{I}_{\mathcal{K}^i}(\cdot)$, and $g(\cdot) \leftarrow \mathcal{I}_{R_+^n}(\cdot + l)$. Detailed discussions specific to our problem are provided as follows.

1) *Dual Update*: In the dual consensus ADMM algorithm, variables $\{y^i\}$ and $\{z^i\}$ are dual variables to the original optimization problem. The updates of $\{y^i\}$ and $\{z^i\}$ are performed by Steps 5-7 and 9-11 of Algorithm 1. In particular, Step 10 is performed by

$$z^{i,k+1,*} = \max\{N(s^{i,k+1} + \sigma y^{i,k+1}), -l + \epsilon\}, \quad (19)$$

which is derived from the definition of the proximal operator. ϵ is a small non-negative number to push z away from the

boundary into the feasible region so as to guarantee strict satisfaction of constraints. $\max\{x, y\}$ denotes the element-wise maximization between two vectors x and y .

2) *Primal Update*: The primal update is performed by Step 8 of Algorithm 1 to search for the optimal states and control inputs for vehicle i . We perform expansion of the square term and obtain

$$\begin{aligned} \delta X^{i,k+1} = \arg \min_{\delta X^i} \{ & C^i(\delta X^i) + \mathcal{I}_{\mathcal{K}^i}(\delta X^i) \\ & + \eta(\delta X^{i\top} J^{i\top} J^i \delta X^i + 2r^{i,k+1\top} J^i \delta X^i) \}. \end{aligned} \quad (20)$$

We perform further expansion of the third term, and the resulting collision avoidance term is

$$\begin{aligned} C_c^i(\delta X^i) &= \eta \sum_{\tau=0}^T \delta x_\tau^{i\top} P_{c,\tau}^i \delta x_\tau^i + 2\Gamma_{c,\tau}^i \delta x_\tau^i, \\ P_{c,\tau}^i &= \sum_{j,\beta,\gamma} J_{\beta,\tau}^{i\top} \bar{n}_{\beta\gamma,\tau}^{ij} \bar{n}_{\beta\gamma,\tau}^{ij\top} J_{\beta,\tau}^i, \\ \Gamma_{c,\tau}^i &= \sum_{j,\beta,\gamma} r_{c,\beta\gamma,j,\tau}^{i,k+1} \bar{n}_{\beta\gamma,\tau}^{ij\top} J_{\beta,\tau}^i, \forall j \neq i, \beta, \gamma \in \{f, r\}. \end{aligned} \quad (21)$$

The resulting road boundary term is

$$\begin{aligned} C_b^i(\delta X^i) &= \eta \sum_{\tau=0}^T \delta x_\tau^{i\top} P_{b,\tau}^i \delta x_\tau^i + 2\Gamma_{b,\tau}^i \delta x_\tau^i, \\ P_{b,\tau}^i &= \sum_{\beta \in \{f, r\}} J_{\beta,\tau}^{i\top} J_{\beta,\tau}^i, \Gamma_{b,\tau}^i = \sum_{\beta \in \{f, r\}} r_{b,\beta,\tau}^{i,k+1} \bar{n}_{\beta,\tau}^{i\top} J_{\beta,\tau}^i. \end{aligned} \quad (22)$$

The resulting control input term is

$$C_{ci}^i(\delta X^i) = \eta \sum_{\tau=0}^{T-1} \delta u_\tau^{i\top} \delta u_\tau^i + 2r_{ci,\tau}^{i,k+1\top} \delta u_\tau^i. \quad (23)$$

In the above equations, $r_{c,\beta\gamma,j,\tau}^{i,k+1}$ is the element of $r^{i,k+1}$ that locates at the same row as the collision avoidance constraint between β circle of vehicle i and γ circle of vehicle j at timestamp τ . $r_{b,\beta,\tau}^{i,k+1}$ and $r_{ci,\tau}^{i,k+1}$ are defined similarly for road boundary constraints and control input constraints, respectively. With the above terms, (20) can be rewritten as

$$\begin{aligned} \min_{\delta X^i} & C^i(\delta X^i) + C_c^i(\delta X^i) + C_b^i(\delta X^i) + C_{ci}^i(\delta X^i) \\ \text{s.t.} & \delta x_{\tau+1}^i = A_\tau^i \delta x_\tau^i + B_\tau^i \delta u_\tau^i. \end{aligned} \quad (24)$$

Obviously, (24) is a standard LQR problem, which can be solved by dynamic programming effectively. Note that (24) only involves state variables and control inputs confined to one vehicle, and therefore (24) corresponding to different vehicles can be solved in a parallel manner.

With the above discussions, we propose Algorithm 2 for solving problem (7). Convex reformulation of problem (7) and loops of dual consensus ADMM are performed in an alternating manner until the termination criterion is met. The update of trajectory is performed following the same method in [23]. Given the convexity of (17), ADMM is guaranteed to converge, and thus consensus will be established after a sufficiently large number of iterations. In practice, we let the algorithm terminate when the trajectories are completely

Algorithm 2 Parallel Optimization Algorithm for Cooperative Driving with Hard Safety Constraints

```

1: for  $i \leftarrow 1$  to  $N$  in parallel:
2:   initialize  $\{x_\tau^i, u_\tau^i\}_{\tau=0}^T, \{p^{i,0}, y^{i,0}, z^{i,0}, s^{i,0}\}, \sigma, \rho$ 
3:   repeat:
4:     Send  $\{x_\tau^i\}_{\tau=1}^T$  to  $j \in \mathcal{N} - \{i\}$ 
5:     Compute  $\{J_{f,\tau}^i, J_{r,\tau}^i, P_{f,\tau}^i, P_{r,\tau}^i\}_{\tau=0}^T, \{A_\tau^i, B_\tau^i\}_{\tau=0}^{T-1}$ 
6:     for  $\tau \leftarrow 0$  to  $T$ :
7:       for  $j$  in  $\mathcal{N} - \{i\}$ :
8:         Compute  $\{p_{f,\tau}^j, p_{r,\tau}^j\}$ 
9:         for  $\beta, \gamma$  in  $\{f, r\}$ :
10:           $\bar{n}_{\beta\gamma,\tau}^{ij} \leftarrow (p_{\beta,\tau}^i - p_{\beta,\tau}^j) / \|p_{\beta,\tau}^i - p_{\beta,\tau}^j\|_2$ 
11:        end
12:       end
13:       for  $\beta$  in  $\{f, r\}$ :
14:          $p_{\beta b,\tau}^i \leftarrow \text{NearestNeighbor}(\mathcal{M}, p_{\beta,\tau}^i)$ 
15:          $\bar{n}_{\beta b,\tau}^i \leftarrow (p_{\beta,\tau}^i - p_{\beta b,\tau}^i) / \|p_{\beta,\tau}^i - p_{\beta b,\tau}^i\|_2$ 
16:       end
17:        $p_{r,ref}^i \leftarrow \text{NearestNeighbor}(\{p_{ref}^i\}, p_\tau^i)$ 
18:       Compute  $Q_\tau^i$ 
19:     end
20:     Compute  $\{P_{c,\tau}^i, P_{b,\tau}^i\}_{\tau=0}^T, J^i, l$ 
21:      $p^{i,0} \leftarrow 0, s^{i,0} \leftarrow 0, y^{i,0} \leftarrow y^{prev}, z^{i,0} \leftarrow z^{prev}$ 
22:     for  $k \leftarrow 0$  to  $k_{\max}$ :
23:       Steps 4-7 of Algorithm 1
24:       Compute  $\{\Gamma_{c,\tau}^i, \Gamma_{b,\tau}^i\}_{\tau=0}^T$ 
25:        $\delta X^{i,k+1} \leftarrow \text{Problem (24)}$ 
26:        $y^{i,k+1} \leftarrow 2\eta(J^i \delta X^{i,k+1} + r^{i,k+1})$ 
27:        $z^{i,k+1} \leftarrow \frac{s^{i,k+1}}{\sigma} + y^{i,k+1}$ 
28:          $-\frac{1}{N\sigma} \max\{N(s^{i,k+1} + \sigma y^{i,k+1}), -l + \epsilon\}$ 
29:     end
30:      $y^{prev} \leftarrow y^{i,k+1}, z^{prev} \leftarrow z^{i,k+1}$ 
31:     Update  $\{x_\tau^i, u_\tau^i\}_{\tau=0}^T$ 
32:   until termination criterion is satisfied
33: end

```

collision-free and the change in the overall cost between two consecutive iterations is smaller than a given criterion ζ .

V. EXPERIMENTAL RESULTS

To verify the proposed method, we perform simulations on the roundabout area in Town03, CARLA simulator [25]. Altogether 16 CAVs are involved in this scenario. For all CAVs, reference paths are pre-generated by the global navigation module in CARLA. The reference paths are identical for all methods and fixed throughout the entire planning process. Key parameter settings are shown in Table I, where d_f and d_r are the biases between the rear axle center and centers of FC and RC . Initial velocities are 10 m/s for all CAVs.

A. Computation Efficiency

To demonstrate the improved performance in terms of computation efficiency, we compare the computation time of our method with the general IPOPT solver provided by CasADi [26]. Both our method and the IPOPT solver are implemented on a server with $2 \times$ Intel(R) Xeon(R) Gold

TABLE I
PARAMETERS USED IN EXPERIMENTS

Param.	Value	Param.	Value	Param.	Value
b	3.0 m	v_{ref}	10 m/s	τ_s	0.1 s
a_{max}	8.0 m/s ²	σ	0.2	ρ	0.02
a_{min}	-12.0 m/s ²	δ_{max}	0.62 rad	ϵ	0.3
k_{max}	2	δ_{min}	-0.62 rad	ζ	1.0
d_{safe}	2.62 m	T	75	d_f	2.79 m
d_r	-0.05 m				

TABLE II
COMPUTATION TIME PER TIME STAMP WITH DIFFERENT METHODS

	Our Method		IPOPT solver	
	Single-process	Multi-process	Two-stage	One-stage
$N=8$	0.00772 s	0.00156 s	0.109 s	0.153 s
$N=12$	0.0235 s	0.00411 s	0.235 s	4.37 s
$N=16$	0.0480 s	0.00876 s	0.506 s	7.09 s

6348 CPU @ 2.60 GHz. For our method, both single-process and multi-process implementations are provided. For the latter one, the number of processes is set to be N .

One difficulty in solving (7) with IPOPT solver is that the reference points need to be defined during the formulation of the problem, which are unknown at the beginning. We simplify (7) for the IPOPT solver by first solving it using our method, and then formulating the problem with the optimal reference points obtained. Meanwhile, we examine two different solving schemes. The one-stage scheme directly optimizes (7) with full constraints, while the two-stage scheme first optimizes (7) with collision avoidance constraints released, and then uses the obtained results as initialization for optimization of the whole problem.

Table II compares the computational efficiency between all four optimization settings. We use the computation time per timestamp as a measure. It can be seen that the multi-process implementation of our method is the fastest of all, which is at least two orders of magnitude faster than the IPOPT solver. This result clearly demonstrates the superiority in computational efficiency of our method for solving cooperative trajectory planning problems. Meanwhile, the multi-process implementation of our method is 4.9 \times , 5.7 \times , and 5.5 \times as fast as the single process implementation for $N = 8, 12$, and 16, respectively. These results demonstrate the effectiveness of parallel optimization.

B. Traffic Efficiency in Roundabout Area

To demonstrate the overall passing efficiency, we compare our method with the baseline proposed by [25], which tracks the reference path using a low-level controller and brakes to maintain safe distance with surrounding vehicles. Qualitative results in Fig. 3 show the positions of vehicles and past trajectories at selected timestamps for our method and the baseline, respectively. Following the given reference paths, smooth and efficient merging into the inner lane is observed in the trajectories generated by our method, which reduces the overall travel distances. Meanwhile, collisions between vehicles are avoided. On the contrary, the baseline exhibits low efficiency in passing and fails to avoid collisions.

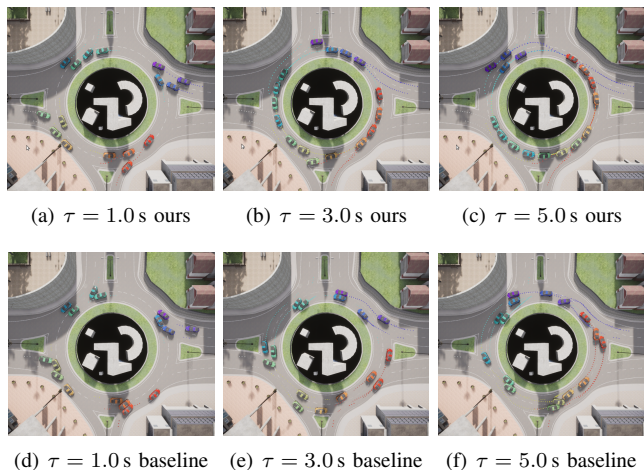


Fig. 3. Simulation results of our method and baseline at time equal to 1.0s, 3.0s, and 5.0s, respectively.

TABLE III
AVERAGE VELOCITY OF VEHICLES WITH DIFFERENT METHODS

		Group 1	Group 2	Group 3	Group 4
$N=8$	Baseline	7.35 m/s	7.81 m/s	7.80 m/s	7.65 m/s
	Proposed	10.0 m/s	9.14 m/s	9.85 m/s	9.33 m/s
$N=12$	Baseline	7.49 m/s	7.74 m/s	6.92 m/s	7.53 m/s
	Proposed	10.0 m/s	9.36 m/s	9.27 m/s	9.40 m/s
$N=16$	Baseline	7.25 m/s	5.51 m/s	6.37 m/s	7.21 m/s
	Proposed	9.86 m/s	9.39 m/s	9.14 m/s	9.08 m/s

Quantitatively, we compare the resulting average velocities of vehicles by our method and the baseline. The vehicles are naturally divided into 4 groups according to the entrances they start from. For each group of vehicles, we compute their average velocities over the entire horizon. The results are shown in Table III. It is clear that for all cases, all 4 groups of vehicles run at higher average velocities with our method than that with the baseline. The average velocities of vehicles attained by our method are close to the reference velocity $v_{ref} = 10$ m/s, while the slowdown is observed in the baseline. These results show that our method yields higher overall passing efficiency for this crowded roundabout area.

VI. CONCLUSIONS

In this paper, we investigate the cooperative autonomous driving problem of CAVs in urban traffic scenarios. We formulate the problem as an optimal control problem with safety conditions enforced as hard constraints, together with other constraints including vehicle dynamics and control limitations. Then, a parallel optimization framework is presented to solve the specific problem efficiently. We compare our method with different baselines to demonstrate the effectiveness in terms of average travel efficiency and to show the superiority in computational efficiency. Possible future works include the incorporation of a high-level decision-making strategy into the optimization framework to further enhance traffic efficiency. We also consider extending the optimization method to a mixed traffic flow that contains both CAVs and human-driving vehicles.

REFERENCES

- [1] R. Deng, B. Di, and L. Song, "Cooperative collision avoidance for overtaking maneuvers in cellular V2X-based autonomous driving," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4434–4446, 2019.
- [2] C. Liu, C.-W. Lin, S. Shiraishi, and M. Tomizuka, "Distributed conflict resolution for connected autonomous vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 1, pp. 18–29, 2017.
- [3] P. Hang, C. Lv, C. Huang, Y. Xing, and Z. Hu, "Cooperative decision making of connected automated vehicles at multi-lane merging zone: A coalitional game approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 4, pp. 3829–3841, 2021.
- [4] J. Rios-Torres and A. A. Malikopoulos, "Automated and cooperative vehicle merging at highway on-ramps," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 4, pp. 780–789, 2016.
- [5] J. Ding, L. Li, H. Peng, and Y. Zhang, "A rule-based cooperative merging strategy for connected and automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 8, pp. 3436–3446, 2019.
- [6] H. Xu, S. Feng, Y. Zhang, and L. Li, "A grouping-based cooperative driving strategy for CAVs merging problems," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, pp. 6125–6136, 2019.
- [7] A. Mirheli, M. Tajalli, L. Hajbabai, and A. Hajbabaie, "A consensus-based distributed trajectory control in a signal-free intersection," *Transportation Research Part C: Emerging Technologies*, vol. 100, pp. 161–176, 2019.
- [8] S. A. Fayazi and A. Vahidi, "Mixed-integer linear programming for optimal scheduling of autonomous vehicle intersection crossing," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 3, pp. 287–299, 2018.
- [9] J. Chen, B. Yuan, and M. Tomizuka, "Model-free deep reinforcement learning for urban autonomous driving," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 2765–2771.
- [10] Y. Guan, Y. Ren, S. E. Li, Q. Sun, L. Luo, and K. Li, "Centralized cooperation for connected and automated vehicles at intersections by proximal policy optimization," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 12 597–12 608, 2020.
- [11] B. Toghi, R. Valiente, D. Sadigh, R. Pedarsani, and Y. P. Fallah, "Cooperative autonomous vehicles that sympathize with human drivers," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 4517–4524.
- [12] X. Pan, B. Chen, S. Timotheou, and S. A. Evangelou, "A convex optimal control framework for autonomous vehicle intersection crossing," *arXiv preprint arXiv:2203.16870*, 2022.
- [13] H. Xu, Y. Zhang, L. Li, and W. Li, "Cooperative driving at unsignalized intersections using tree search," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4563–4571, 2019.
- [14] T. Kessler and A. Knoll, "Cooperative multi-vehicle behavior coordination for autonomous driving," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 1953–1960.
- [15] K. Esterle, T. Kessler, and A. Knoll, "Optimal behavior planning for autonomous driving: A generic mixed-integer formulation," in *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2020, pp. 1914–1921.
- [16] C. Burger and M. Lauer, "Cooperative multiple vehicle trajectory planning using MIQP," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 602–607.
- [17] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. a. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [18] F. Rey, Z. Pan, A. Hauswirth, and J. Lygeros, "Fully decentralized ADMM for coordination and collision avoidance," in *2018 European Control Conference (ECC)*. IEEE, 2018, pp. 825–830.
- [19] Z. Wang, Y. Zheng, S. E. Li, K. You, and K. Li, "Parallel optimal control for cooperative automation of large-scale connected vehicles via ADMM," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 1633–1639.
- [20] X. Zhang, Z. Cheng, J. Ma, S. Huang, F. L. Lewis, and T. H. Lee, "Semi-definite relaxation-based ADMM for cooperative planning and control of connected autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 9240–9251, 2021.
- [21] A. D. Saravanos, Y. Aoyama, H. Zhu, and E. A. Theodorou, "Distributed differential dynamic programming architectures for large-scale multi-agent control," *arXiv preprint arXiv:2207.13255*, 2022.
- [22] G. Banjac, F. Rey, P. Goulart, and J. Lygeros, "Decentralized resource allocation via dual consensus ADMM," in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 2789–2794.
- [23] Z. Huang, S. Shen, and J. Ma, "Decentralized iLQR for cooperative trajectory planning of connected autonomous vehicles via dual consensus ADMM," *arXiv preprint arXiv:2301.04386*, 2023.
- [24] Y. Tassa, N. Mansard, and E. Todorov, "Control-limited differential dynamic programming," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 1168–1175.
- [25] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Conference on Robot Learning*. PMLR, 2017, pp. 1–16.
- [26] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, pp. 1–36, 2019.