arXiv:2303.03090v2 [cs.RO] 4 May 2024

面向网联自动驾驶车辆协同规划的具有硬安全约束的并行优化

Zhenmin Huang、Haichao Liu、Shaojie Shen 和 Jun Ma

摘要—网联自动驾驶车辆（CAVs）的发展有助于提升复杂场景下的交通效率。然而，如何为网联自动驾驶车辆制定有效且高效的协调策略，仍有待解决。本文将网联自动驾驶车辆的协同自动驾驶任务建模为一个带有硬安全约束的最优控制问题，并提出一种计算高效的并行优化框架，用于生成车辆策略，在满足硬安全约束的同时提升通行效率。具体而言，我们通过凸近似妥善处理所有相关约束，使重构后的优化问题具有凸性。随后，提出一种并行优化算法来求解重构后的优化问题，并结合迭代最近邻搜索策略确定最优通行顺序。值得注意的是，采用所提出的创新方案后，通行效率得到提升，计算负担也大幅减轻。我们还在 CARLA 仿真器中检验了所提出的方法，并进行了全面对比，以验证该方法的有效性和效率。

I. 引言

人工智能和信息技术的最新发展推动了网联自动驾驶车辆（CAVs）的出现。这类车辆能够通过 5G 网络和 Wi-Fi 等实时通信技术在车辆之间交换信息 [1]。在此意义上，协同自动驾驶得以实现，有望缓解由非协同驾驶方式引发的交通拥堵，并减少交通事故 [2]。尽管其物理基础已经具备，但仍有一系列问题尚未得到充分解决

本研究得到中国国家自然科学基金（项目编号：62303390）的部分资助；以及河套深港科技创新合作区项目（项目编号：HZQB-KCZYB-2020083）的部分资助。（通讯作者：*Jun Ma*。）

Zhenmin Huang 就职于中国广州的香港科技大学（广州）机器人与自主系统方向，同时就职于中国香港特别行政区的香港科技大学新兴跨学科领域学部（电子邮箱：zhuangdf@connect.ust.hk）。

Haichao Liu 任职于中国广州香港科技大学（广州）机器人与自主系统方向（电子邮箱：hliu369@connect.hkust-gz.edu.cn）。

Shaojie Shen 任职于中国香港特别行政区香港科技大学电子与计算机工程系（电子邮箱：eeshaojie@ust.hk）。

Jun Ma 任职于中国广州的香港科技大学（广州）机器人与自主系统学部，同时任职于中国香港特别行政区香港科技大学新兴跨学科领域学部，以及中国深圳福田的港科大深港协同创新研究院（电子邮箱：jun.ma@ust.hk）。

在开发适用于复杂城市交通场景下协同自动驾驶的成熟协调方案方面仍未取得突破。针对不同的城市交通场景（如匝道合流 [3], [4], [5],[6] 和交叉路口通行 [7], [8]），研究人员深入研究了基于规则、基于优化和基于学习的方法，以制定各种协调策略。然而，现有方法要么仅关注简单场景，要么只考虑数量有限的网联自动驾驶车辆之间的简单交互，因此难以直接应用于交通流密集的复杂城市交通场景。此外，现有方法普遍存在计算效率低的问题。随着场景中网联自动驾驶车辆数量增加，问题规模不断扩大，导致整体计算时间迅速增长，无法满足网联自动驾驶车辆规划与控制的实时要求。因此，当前方法通常无法应用于包含大量网联自动驾驶车辆的场景。

本文聚焦于开发一种优化框架，以解决城市交通场景中全方向交通流的协同自动驾驶问题。为提供严格的安全保障，本文引入一种线性化方法，以妥善处理复杂的非线性碰撞避免约束和道路边界约束。为提高通行效率，本文提出一种基于迭代最近邻搜索的新型协调策略，用于确定最优通行顺序。结合基于对偶共识ADMM的并行优化方法，该方法大幅减轻了计算负担，从而实现实时运行。在实验验证中，我们将我们的方法与若干基线方法进行比较，以展示其在提高通行效率和计算效率方面的优势。

II. 相关工作

针对城市交通场景中的协同驾驶策略，人们进行了大量研究。得益于数据科学和人工智能的迅速发展，基于学习的方法目前突飞猛进，并利用真实世界数据提升性能。有研究指出，基于学习的方法能够有效应对复杂的交通场景 [9]。在 [10],中，提出了一种基于强化学习的集中式协调方案，可实现网联自动驾驶车辆在无信号交叉路口的协同通行。在 [11],中，采用多智能体强化学习，使网联自动驾驶车辆能够与人类驾驶员协同，从而高效地

汇入高速公路。尽管基于学习的方法潜力巨大，但通常需要大量真实世界数据，而且可能存在可解释性较差的问题，这阻碍了其在真实世界场景中的广泛应用。

与基于学习的方法相比，基于规则和基于优化的方法更加成熟，且能够在计算资源有限的情况下生成可预测、可解释的结果。具体而言，基于规则的方法主要利用人工制定的规则和准则来确定网联自动驾驶车辆的通行顺序，然后由低层规划器为所有网联自动驾驶车辆生成轨迹。在 [2], 中，提出了一种通信支持的分布式机制，用于解决交叉口中网联自动驾驶车辆之间的冲突。每辆车在本地求解冲突图，以计算期望的通行时间段，随后由运动规划器确定速度曲线。在 [12], 中，提出了一种集中式凸最优控制框架，并采用分层协调方案，使网联自动驾驶车辆能够通过无信号交叉路口。在 [13],中，提出了一种基于蒙特卡洛树搜索（MCTS）和启发式规则的方法，以寻找网联自动驾驶车辆通过无信号交叉口的近似全局最优顺序。尽管基于规则的方法简单直接，但人工制定的规则通常难以涵盖各种边缘情况，导致这类方法难以泛化到复杂场景。

与此同时，基于优化的方法通常将协同驾驶建模为约束优化问题，并采用统一优化框架为所有网联自动驾驶车辆直接进行规划，因此表达更简洁且更具泛化能力 [14], [15]。在 [16], 中，提出了一种基于混合整数二次规划（MIQP）的方法，以求解一般道路场景下网联自动驾驶车辆协同驾驶的全局最优解。在 [7],中，协同轨迹规划问题被建模为混合整数非线性规划（MINLP），以确定无冲突的轨迹。然而，这些方法仅适用于少量车辆，因为车辆数量增多时，计算时间很快就会变得难以承受。为提高计算效率，亟需采用分布式并行优化方法。交替方向乘子法（ADMM） [17] 提供了一种通用的分布式框架，可加速优化过程，因此近年来开始应用于协同自动驾驶 [18]。在 [19], [20],中，ADMM用于将独立的车辆约束与车辆间耦合的碰撞规避约束分离开来，以便分别处理，从而形成部分并行的优化方案。在 [21],中，提出了一种完全分布式并行优化框架，用于求解一般类型机器人的协同轨迹规划问题。然而，其计算效率仍不尽如人意，仅涉及数个机器人的问题也需要数分钟才能求解。基于对偶共识ADMM， [22], 中提出了一种并行优化框架， [23],该框架能够以较高的计算效率实现网联自动驾驶车辆的协同轨迹规划。然而，该方法采用软惩罚来实现碰撞规避，若惩罚设置不当，可能导致碰撞或过于保守的策略。此外，该方法忽略了某些与驾驶安全相关的重要约束，例如道路边界限制。这些问题使得生成的策略在现实场景中可能并不安全。此外，所有网联自动驾驶车辆的时间进度均以非协同方式预先指定且固定，这可能导致难以解决车道汇入等场景中的冲突。

III. 问题陈述

A. 车辆运动学与物理约束

对于涉及 N 辆网联自动驾驶车辆同时通过的交通场景，我们用 $\mathcal{N} = \{1, 2, ..., N\}$ 表示包含所有网联自动驾驶车辆索引的集合。假设通信网络具有完全图结构，且所有参与其中的网联自动驾驶车辆具有相同的运动学特性，可表示为

$$x^i_{\tau+1} = f(x^i_\tau, u^i_\tau), \tag{1}$$

其中， x^i_τ 和 u^i_τ 分别表示车辆 i 在时间戳 τ处的状态向量和控制输入向量， $\tau \in \mathcal{T} =\{0, 1, ..., T-1\}$，且 T 为时域长度。具体而言，我们考虑由四个状态变量$x^i_\tau = (p^i_{x,\,\tau}$， $p^i_{y,\,\tau}$， θ^i_τ， $v^i_\tau)$组成的状态向量，以及由两个控制输入 $u^i_\tau =(\delta^i_\tau$， $a^i_\tau)$构成的输入向量，其中 $p^i_{x,\,\tau}$和 $p^i_{y,\,\tau}$分别表示全局坐标系中后轴中点的 X 坐标和 Y 坐标， θ^i_τ 表示航向角， v^i_τ表示速度， δ^i_τ 表示转向角，a^i_τ 表示加速度。根据以上记号，我们采用以下车辆 i的运动学模型 [24]：

$$\begin{cases} p^i_{x,\tau+1} = p^i_{x,\tau} + f_r(v^i_\tau, \delta^i_\tau)\cos(\theta^i_\tau), \\ p^i_{y,\tau+1} = p^i_{y,\tau} + f_r(v^i_\tau, \delta^i_\tau)\sin(\theta^i_\tau), \\ \theta^i_{\tau+1} = \theta^i_\tau + \arcsin\left(\tau_s v^i_\tau \sin(\delta^i_\tau)/b\right), \\ v^i_{\tau+1} = v^i_\tau + \tau_s a^i_\tau. \end{cases} \tag{2}$$

注意， b 是车辆轴距， τ_s 是时间间隔。函数 $f_r(v,\ \delta)$ 定义为

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

同时，对控制输入施加以下约束:

$$a_{\min} \le a^i_\tau \le a_{\max},\ \delta_{\min} \le \delta^i_\tau \le \delta_{\max}. \tag{4}$$

B. 避碰约束

为确保驾驶安全，必须对网联自动驾驶车辆之间的最小距离施加约束，以避免任何可能的碰撞。我们假设所有网联自动驾驶车辆的尺寸均相同，并将每辆车辆近似为沿纵向排列的两个相同圆。车辆 i 的前圆和后圆分别记为 FC_i 和 RC_i。在这种近似下，避免碰撞的充分条件是：不同车辆上任意一对圆的中心之间的距离不小于直径。我们将其中心位置记为

在时间 τ，FC_i 和 RC_i 的中心位置分别为 $p^i_{f,\tau}$ 和 $p^i_{r,\tau}$。因此，无碰撞条件可表示为

$$||p^i_{\beta,\tau} - p^j_{\gamma,\tau}||_2 \geq d_{\text{safe}}. \tag{5}$$

注意，d_{safe} 是需要保持的安全距离，该距离被设定为所有圆的共同直径。

C. 道路边界约束

为确保所有网联自动驾驶车辆均在自由空间内行驶，必须施加道路边界约束。满足道路边界约束的一个充分条件是：圆心与相应最近道路边界之间的最小距离始终不小于所有圆的半径。等价地，我们将所有道路边界向自由空间内部移动一段偏移量，该偏移量等于所有圆的共同半径。由此形成了一个新的集合 $\mathcal{C}$，它是自由空间的子集。随后建立以下约束：

$$p^i_{\beta,\tau} \in \mathcal{C}. \tag{6}$$

D. 总体公式化

对于网联自动驾驶车辆，我们将目标设定为最小化实际轨迹与参考轨迹之间的偏差，以及控制输入。综合上述所有约束，我们得到如下最优控制问题：

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

$x^i_{\tau,ref}$ 是车辆 i 在时间戳 τ处的参考点，其中包含参考航点、参考航向角和参考速度。$||\cdot||_Q$ 和 $||\cdot||_R$ 分别表示由矩阵 Q 和 R加权的 L_2范数，其中 Q 为半正定矩阵，R 为正定矩阵。

IV. 方法

本节提出一个优化框架，用于求解问题 (7)，以获得最优协同驾驶策略。稍微滥用一下符号，我们将当前的名义轨迹定义为 $\{x^i_\tau, u^i_\tau\}$，分别对应 $\tau \in \mathcal{T}$和 $i \in \mathcal{N}$。我们的目标是在名义轨迹上搜索最优变化量 $\{\delta x^i_\tau, \delta u^i_\tau\}$，使目标函数最小化，同时满足约束。该过程需要反复迭代，直至所有网联自动驾驶车辆的轨迹最终收敛。

A. 硬安全约束的线性化

城市交通场景中的安全约束包括避碰约束和道路边界约束。需要注意的是，这两类约束均施加于 FC 和 RC的中心位置，而优化变量则绑定于网联自动驾驶车辆的后轴中心。这一差异需要通过中间的雅可比矩阵加以衔接，即

$$\delta p^i_{\beta,\tau} = J^i_{\beta,\tau} \delta x^i_\tau. \tag{8}$$

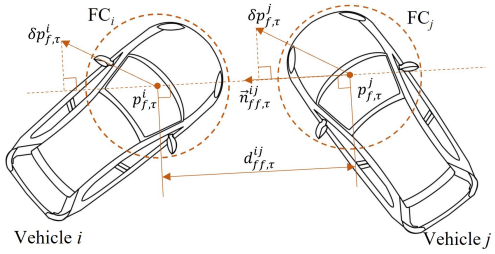


图 1. 避碰约束的线性化。

原始的碰撞避免约束显然是非凸且非线性的，这可能导致难以找到可行解。与 [18],类似，我们采用一种直接的方法将这些约束线性化。对于车辆 i 与车辆 j之间的每一对圆，我们构造一个单位向量$\vec{n}^{ij}_{\beta\gamma,\tau}$ 该向量从一个圆的中心指向另一个圆的中心。随后，不等式提供了这两个圆之间实现碰撞规避的充分条件：

$$\vec{n}^{ij,\top}_{\beta\gamma,\tau}(J^i_{\beta,\tau}\delta x^i_\tau - J^j_{\gamma,\tau}\delta x^j_\tau) + d^{ij}_{\beta\gamma,\tau} - d_{\text{safe}} \geq 0. \tag{9}$$

该不等式背后的原理是，$\delta p^i_{\beta,\tau}$ 和 $\delta p^j_{\gamma,\tau}$ 在 $\vec{n}^{ij}_{\beta\gamma,\tau}$ 上的投影之差，给出了当前距离变化量 $d^{ij}_{\beta\gamma,\tau}$的下界，如图 1 所示。因此，当该差值与当前距离之和大于或等于 d_{safe} 时，即满足无碰撞条件。

为处理不规则道路边界，我们提出了一种线性化方法，理论上可应用于任意形状的道路边界约束。我们首先提取场景中的所有道路边界，然后对边界进行密集采样，将其转换为点云，记为$\mathcal{M}$。随后，对 $\mathcal{M}$执行最近邻搜索，以获取每个时间戳处与车辆位置对应的最近边界点：

$$p^i_{\beta b,\tau} \leftarrow NearestNeighbor(\mathcal{M}, p^i_{\beta,\tau}), \tag{10}$$

其中，$p^i_{\beta b,\tau}$是$p^i_{\beta,\tau}$中距离 $\mathcal{M}$最近的点。然后，我们构造从$p^i_{\beta b,\tau}$指向$p^i_{\beta,\tau}$的单位向量，记为$\vec{n}^i_{\beta,\tau}$。按照与式（9）相同的逻辑，线性化道路边界约束可表示为

$$2\vec{n}^{i,\top}_{\beta,\tau} J^i_{\beta,\tau} \delta x^i_\tau + 2d^i_{\beta,\tau} - d_{\text{safe}} \geq 0. \tag{11}$$

图2展示了线性化道路边界约束。为高效获取最近点，我们使用k-d树执行最近邻搜索；对于包含 N 个点的点云，单次查询的复杂度为$O(logN)$。

B. 隐式通行顺序协调

在城市交通场景中，两辆车辆试图同时并入同一车道是一种常见情况。这带来了确定通行顺序这一关键问题。在本部分中，我们提出了一种新颖策略，通过迭代构造目标函数，隐式地获得最优通行顺序。我们首先定义

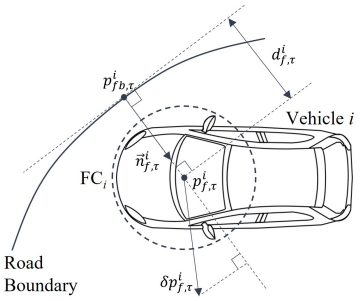


图 2. 道路边界约束的线性化。

$\delta X^i = (\delta x^i_0,\ \delta u^i_0,\ \ldots\ldots \delta x^i_T)$ 为与车辆 i 对应的所有变分状态和输入所组成的连接向量。根据（7），与车辆 i 对应的目标函数为

$$\begin{aligned} C^i(\delta X^i) = &\sum_{\tau=0}^{T} ||\delta x^i_\tau||^2_{Q^i_\tau} + 2(x^i_\tau - x^i_{\tau,ref})^\top Q^i_\tau \delta x^i_\tau \\ &+ \sum_{\tau=0}^{T-1} ||\delta u^i_\tau||^2_R + 2u^{i\top}_\tau R\delta u^i_\tau. \end{aligned} \tag{12}$$

由于车辆 i 的通行顺序尚未确定，因此时间戳 τ 对应的参考点$x^i_{\tau,\ ref}$ 未知。为获得最优参考航点，我们采用最近邻搜索。假设导航模块为车辆 i 生成的参考路径为 $\{p^i_{ref}\}$，即该路径所包含的所有航路点的集合。时间 τ 的当前参考航点表示为

$$p^i_{\tau,ref} \leftarrow NearestNeighbor(\{p^i_{ref}\}, p^i_\tau), \tag{13}$$

其中 $p^i_\tau = (p^i_{x,\ \tau},\ p^i_{y,\ \tau})$。在每次优化迭代中，都会执行最近邻搜索并重构目标函数。因此，已经领先的车辆会将自己的参考航路点设得更靠前，而落后的车辆则相反。结合安全约束，各车辆会在迭代过程中逐渐分开；最终确定通行顺序。

为提高效率，最近航路点的搜索也采用 k-d树实现。所有车辆在所有时间戳均使用恒定参考速度 v_{ref}，且不限制航向角。此外，协方差矩阵 Q^i_τ 的定义仅考虑相对于参考路径的横向偏差，而忽略沿参考路径相对于参考航点的纵向偏差。

C. 优化问题重构

为处理非线性运动学，我们对式 (1) 进行一阶泰勒展开，得到

$$\delta x^i_{\tau+1} = A^i_\tau \delta x^i_\tau + B^i_\tau \delta u^i_\tau. \tag{14}$$

输入约束 (15) 被重新表述为

$$a_{\min} - a^i_\tau \le \delta a^i_\tau \le a_{\max} - a^i_\tau, \delta_{\min} - \delta^i_\tau \le \delta\delta^i_\tau \le \delta_{\max} - \delta^i_\tau. \tag{15}$$

Algorithm 1 Dual Consensus ADMM [22]

choose $\sigma, \rho > 0$
initialize for all $i \in \mathcal{N}$: $p^{i,0} = y^{i,0} = z^{i,0} = s^{i,0} = 0$
repeat: for all $i \in \mathcal{N}$
  Broadcast $y^{i,k}$ to all other vehicles
  $p^{i,k+1} \leftarrow p^{i,k} + \rho \sum_{j\neq i}(y^{i,k} - y^{j,k})$
  $s^{i,k+1} \leftarrow s^{i,k} + \sigma(y^{i,k} - z^{i,k})$
  $r^{i,k+1} \leftarrow \rho \sum_{j\neq i}(y^{i,k} + y^{j,k}) + \sigma z^{i,k} - p^{i,k+1} - s^{i,k+1}$
  $x^{i,k+1} \leftarrow \arg\min_{x^i}\{f^i(x^i) + \eta||J^i x^i + r^{i,k+1}||^2\}$
  $y^{i,k+1} \leftarrow 2\eta(J^i x^{i,k+1} + r^{i,k+1})$
  $z^{i,k+1,*} \leftarrow \mathrm{Prox}_g^{\frac{1}{N\sigma}}(N(s^{i,k+1} + \sigma y^{i,k+1}))$
  $z^{i,k+1} \leftarrow \frac{s^{i,k+1}}{\sigma} + y^{i,k+1} - \frac{1}{N\sigma} z^{i,k+1,*}$
until termination criterion is satisfied

我们将 (9)、(11) 和 (15) 展开并改写为紧凑的矩阵形式，得到

$$\sum_{i=1}^{N} J^i \delta X^i + l \ge 0. \tag{16}$$

我们将给定集合 $\mathcal{S}$ 上的指示函数记为 $\mathcal{I}_{\mathcal{S}}(x)$。采用这一记法，将式 (16)、目标函数和线性化运动学汇总，得到以下二次优化问题

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

其中，$\mathcal{K}^i$ 是车辆 i 线性化运动学的可行集，R^n_+ 是非负正交象限，n 为 l 的长度。

D. 并行优化

采用对偶共识ADMM进行并行优化。我们将近端算子定义为 $\mathrm{Prox}^\rho_f(x) = \arg\min_y \{f(y) + \frac{\rho}{2}||x - y||^2\}$，算法 1 [22] 可用于求解以下凸优化问题：

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

注意，$p,\ s,\ r,\ y,\ z$是与 l同尺寸的列向量，且 $\eta = 1/(2(\sigma + 2\rho(N-1)))$。我们通过以下替换将对偶共识ADMM应用于问题(17)：$x^i \leftarrow \delta X^i$, $f^i(\cdot) \leftarrow C^i(\cdot) + \mathcal{I}_{\mathcal{K}^i}(\cdot)$, 以及 $g(\cdot) \leftarrow \mathcal{I}_{R^n_+}(\cdot + l)$。下面将针对我们的问题进行详细讨论。

1）对偶更新：在对偶共识ADMM算法中，变量 $\{y^i\}$ 和 $\{z^i\}$ 是原始优化问题的对偶变量。$\{y^i\}$ 和 $\{z^i\}$的更新由算法1的步骤5–7和9–11完成。具体而言，步骤10通过以下方式执行：

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

它由近端算子的定义推导而来。ϵ 是一个较小的非负数，用于将 z 从

边界推入可行域，以保证约束的严格满足。$\max\{x, y\}$ 表示两个向量 x 和 y之间逐元素取最大值。

2) 原始更新：算法1的第8步执行原始更新，以搜索车辆 i的最优状态和控制输入。我们展开平方项，得到

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

我们进一步展开第三项，得到的碰撞避免项为

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

得到的道路边界项为

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

得到的控制输入项为

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

在上述公式中，$r^{i,\,c\quad,\,\beta\gamma,\,j,\,\tau}$ 是 $r^{i,\,k+1}$ 中的一个元素，该元素位于与时间戳 τ时车辆 i 的 β 圆和车辆 j 的 γ 圆之间的碰撞避免约束相同的行。$r^{i,\,b\quad\tau\quad,\,\beta,}$ 以及 $r^{i,\,ci\,\tau\quad,}$ 分别针对道路边界约束和控制输入约束以类似方式定义。根据上述各项，(20)可改写为

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

显然，(24)是一个标准的线性二次型调节器问题，可通过动态规划有效求解。请注意，(24)仅涉及限定于单辆车的状态变量和控制输入，因此，对不同车辆对应的(24)可以并行求解。

基于上述讨论，我们提出算法 2 来求解问题 (7)。以交替方式执行问题 (7) 的凸重构和对偶共识ADMM 循环，直至满足终止准则。轨迹的更新采用与 [23] 中相同的方法。鉴于 (17) 的凸性，ADMM 保证收敛，因此经过足够多次迭代后将建立共识。在实践中，当轨迹完全

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

for $i \leftarrow 1$ **to** N **in parallel**:
  initialize $\{x_\tau^i, u_\tau^i\}_{\tau=0}^T, \{p^{i,0}, y^{i,0}, z^{i,0}, s^{i,0}\}, \sigma, \rho$
  repeat:
    Send $\{x_\tau^i\}_{\tau=1}^T$ to $j \in \mathcal{N} - \{i\}$
    Compute $\{J_{f,\tau}^i, J_{r,\tau}^i, p_{f,\tau}^i, p_{r,\tau}^i\}_{\tau=0}^T$, $\{A_\tau^i, B_\tau^i\}_{\tau=0}^{T-1}$
    for $\tau \leftarrow 0$ **to** T:
      for j **in** $\mathcal{N} - \{i\}$:
        Compute $\{p_{f,\tau}^j, p_{r,\tau}^j\}$
        for β, γ **in** $\{f, r\}$:
          $\vec{n}_{\beta\gamma,\tau}^{ij} \leftarrow (p_{\beta,\tau}^i - p_{\gamma,\tau}^j)/||p_{\beta,\tau}^i - p_{\gamma,\tau}^j||_2$
        end
      end
      for β **in** $\{f, r\}$:
        $p_{\beta b,\tau}^i \leftarrow NearestNeighbor(\mathcal{M}, p_{\beta,\tau}^i)$
        $\vec{n}_{\beta,\tau}^i \leftarrow (p_{\beta,\tau}^i - p_{\beta b,\tau}^i)/||p_{\beta,\tau}^i - p_{\beta b,\tau}^i||_2$
      end
      $p_{\tau,ref}^i \leftarrow NearestNeighbor(\{p_{ref}^i\}, p_\tau^i)$
      Compute Q_τ^i
    end
    Compute $\{P_{c,\tau}^i, P_{b,\tau}^i\}_{\tau=0}^T, J^i, l$
    $p^{i,0} \leftarrow 0, s^{i,0} \leftarrow 0, y^{i,0} \leftarrow y^{prev}, z^{i,0} \leftarrow z^{prev}$
    for $k \leftarrow 0$ **to** $k_{\max}$:
      Steps 4-7 of **Algorithm 1**
      Compute $\{\Gamma_{c,\tau}^i, \Gamma_{b,\tau}^i\}_{\tau=0}^T$
      $\delta X^{i,k+1} \leftarrow$ Problem (24)
      $y^{i,k+1} \leftarrow 2\eta(J^i \delta X^{i,k+1} + r^{i,k+1})$
      $z^{i,k+1} \leftarrow \frac{s^{i,k+1}}{\sigma} + y^{i,k+1}$
        $-\frac{1}{N\sigma}\max\{N(s^{i,k+1} + \sigma y^{i,k+1}), -l + \epsilon\}$
    end
    $y^{prev} \leftarrow y^{i,k+1}, z^{prev} \leftarrow z^{i,k+1}$
    Update $\{x_\tau^i, u_\tau^i\}_{\tau=0}^T$
  until termination criterion is satisfied
end

无碰撞，且连续两次迭代之间总体代价的变化小于给定准则 ζ时，算法即终止。

V. 实验结果

为验证提出的方法，我们在 CARLA 仿真器的 Town03 环岛区域中进行了仿真 [25]。该场景共涉及 16 辆网联自动驾驶车辆。对于所有网联自动驾驶车辆，参考路径均由 CARLA 中的全局导航模块预先生成。所有方法使用的参考路径均相同，并在整个规划过程中保持不变。关键参数设置如表 I 所示，其中 d_f 和d_r 分别表示后轴中心与FC和 RC中心之间的偏差。所有网联自动驾驶车辆的初始速度均为 10 m/s。

A. 计算效率

为展示计算效率方面的性能提升，我们将我们的方法与 CasADi 提供的通用 IPOPT 求解器的计算时间进行比较 [26]。我们的方法和 IPOPT 求解器均在一台配备 $2\times$ Intel(R) Xeon(R) Gold

表 I 实验中使用的参数

参数	数值	参数	数值	参数	数值
b	3.0 米	v_{ref}	10 米/秒	τ_s	0.1 秒
$a_{\max}$	8.0 米/s^2	σ	0.2	ρ	0.02
$a_{\min}$	-12.0 米/s^2	$\delta_{\max}$	0.62 弧度	ϵ	0.3
$k_{\max}$	2	$\delta_{\min}$	-0.62 弧度	ζ	1.0
d_{safe}	2.62 米	T	75	d_f	2.79 米
d_r	-0.05 米				

表 II 不同方法的每个时间戳的计算时间

	我们的方法		IPOPT 求解器	
	单进程	多进程	两阶段	单阶段
N=8	0.00772 秒	**0.00156 秒**	0.109 秒	0.153 秒
N=12	0.0235 秒	**0.00411 秒**	0.235 秒	4.37 秒
N=16	0.0480 秒	**0.00876 秒**	0.506 秒	7.09 秒

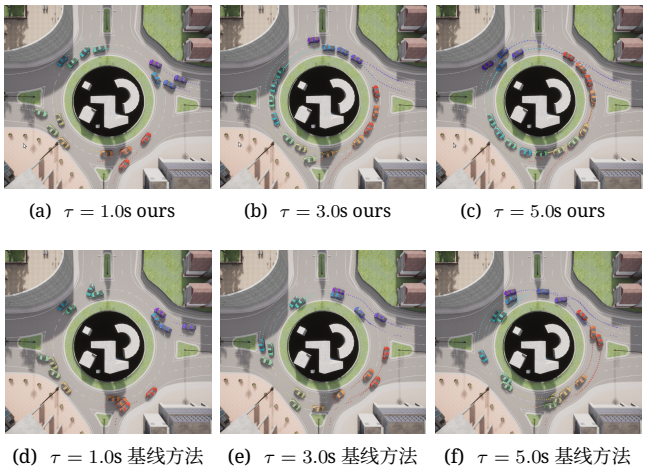
6348CPU @2.60 吉赫兹的服务器上实现。我们的方法提供单进程和多进程两种实现。对于后者，进程数设为 N。

使用 IPOPT 求解器求解 (7) 的一个难点在于，问题构建时需要定义参考点，但这些参考点在初始阶段尚未知晓。我们先用我们的方法求解 (7)，再根据得到的最优参考点构建问题，从而简化 IPOPT 求解器的求解过程。同时，我们考察了两种不同的求解方案。单阶段方案直接在完整约束下优化 (7)，而两阶段方案先在放宽碰撞避免约束的情况下优化 (7)，然后将所得结果作为优化整个问题的初始化值。

表 II 比较了四种优化设置的计算效率。我们以每个时间戳的计算时间作为衡量指标。可以看出，我们的方法的多进程实现速度最快，至少比 IPOPT 求解器快两个数量级。这一结果清楚地表明，我们的方法在求解协同轨迹规划问题时具有更高的计算效率。同时，对于 $N = 8$，12，和 16，我们的方法的多进程实现分别比单进程实现快 $4.9\times$、$5.7\times$ 和 $5.5\times$ 倍。这些结果证明了并行优化的有效性。

B. 环岛区域的交通效率

为了展示整体通行效率，我们将我们的方法与 [25], 提出的基线方法进行比较。该基线方法使用低级控制器跟踪参考路径，并通过制动与周围车辆保持安全距离。图 3 中的定性结果分别展示了我们的方法和基线方法在所选时间戳处的车辆位置及过去的轨迹。按照给定的参考路径，我们的方法生成的轨迹显示，车辆能够平稳高效地并入内车道，从而减少整体行驶距离。同时，车辆之间的碰撞得以避免。相比之下，基线方法通行效率较低，且无法避免碰撞。



(a) $\tau = 1.0$s ours (b) $\tau = 3.0$s ours (c) $\tau = 5.0$s ours

(d) $\tau = 1.0$s 基线方法 (e) $\tau = 3.0$s 基线方法 (f) $\tau = 5.0$s 基线方法

图 3. 我们的方法和基线方法在时间分别为 1.0 s、3.0 s 和 5.0 s 时的仿真结果。

表 III 不同方法下车辆的平均速度

		第1组	第2组	第3组	第4组
N=8	基线方法	7.35 米/秒	7.81 米/秒	7.80 米/秒	7.65 米/秒
	提出的方法	**10.0** 米/秒	**9.14** m/s	**9.85** 米/秒	**9.33** 米/秒
N=12	基线方法	7.49 米/秒	7.74 米/秒	6.92 米/秒	7.53 米/秒
	提出的方法	**10.0** 米/秒	**9.36** 米/秒	**9.27** 米/秒	**9.40** 米/秒
N=16	基线方法	7.25 米/秒	5.51 米/秒	6.37 米/秒	7.21 米/秒
	提出的方法	**9.86** 米/秒	**9.39** m/s	**9.14** 米/秒	**9.08** m/s

定量而言，我们比较了我们的方法与基线方法所得的车辆平均速度。根据车辆出发的入口，可将车辆自然分为 4 组。对于每组车辆，我们计算其在整个时间范围内的平均速度。结果如表 III 所示。显然，在所有情况下，我们的方法下 4 组车辆的平均速度都高于基线方法。我们的方法所达到的车辆平均速度接近参考速度 $v_{\text{ref}} = 10\text{m/s}$，而基线方法则出现了减速现象。这些结果表明，对于这一拥挤的环形交叉口区域，我们的方法能够实现更高的整体通行效率。

六、结论

本文研究了城市交通场景中网联自动驾驶车辆的协同自动驾驶问题。我们将该问题构建为一个最优控制问题，将安全条件作为硬约束强制执行，同时纳入车辆动力学和控制限制等其他约束。随后，提出一种并行优化框架，以高效求解这一具体问题。我们将我们的方法与不同的基线方法进行比较，以展示其在平均通行效率方面的有效性，并证明其计算效率方面的优势。未来的研究可将高层决策策略纳入优化框架，以进一步提升交通效率。我们还考虑将该优化方法扩展到包含

网联自动驾驶车辆和人工驾驶车辆。

参考文献

[1] R. Deng、B. Di 和 L. Song，《基于蜂窝车联网的自动驾驶中超车机动的协同碰撞规避》，*IEEE*车辆技术汇刊，第68卷，第5期，第4434–4446页，2019年。[2] C. Liu、C.-W. Lin、S. Shiraishi 和 M. Tomizuka，《网联自动驾驶汽车的分布式冲突消解》，*IEEE*智能车辆汇刊，第3卷，第1期，第18–29页，2017年。[3] P. Hang、C. Lv、C. Huang、Y. Xing 和 Z. Hu，《多车道汇入区网联自动化车辆的协同决策：一种联盟博弈方法》，*IEEE*智能交通系统汇刊，第23卷，第4期，第3829–3841页，2021年。[4] J. Rios-Torres 和 A. A. Malikopoulos，《高速公路入口匝道处的自动化与协同车辆汇入》，*IEEE*智能交通系统汇刊，第18卷，第4期，第780–789页，2016年。[5] J. Ding、L. Li、H. Peng 和 Y. Zhang，《面向网联自动化车辆的基于规则的协同汇入策略》，*IEEE*智能交通系统汇刊，第21卷，3446，2019 8，3436 no. pp. – .[6] H. Xu、S. Feng、Y. Zhang 和 L. Li，《面向网联自动驾驶车辆汇入问题的基于分组的协同驾驶策略》，*IEEE*车辆技术汇刊，第68卷，第6期，第6125–6136页，2019年。[7] A. Mirheli、M. Tajalli、L. Hajibabai 和 A. Hajbabaie，《无信号交叉口中基于共识的分布式轨迹控制》，《交通研究*C*辑：新兴技术》，第100卷，第161–176页，2019年。[8] S. A. Fayazi 和 A. Vahidi，《用于自动驾驶车辆通过交叉口最优调度的混合整数线性规划》，*IEEE*智能车辆汇刊，第3卷，第3期，第287–299页，2018年。[9] J. Chen、B. Yuan 和 M. Tomizuka，《城市自动驾驶的无模型深度强化学习》，载于*2019*年*IEEE*智能交通系统会议（*ITSC*）。IEEE，2019年，第2765–2771页。[10] Y. Guan、Y. Ren、S. E. Li、Q. Sun、L. Luo 和 K. Li，《通过近端策略优化实现交叉口网联自动化车辆的集中式协同》，*IEEE*车辆技术汇刊，第69卷，第11期，第12 597–12 608页，2020年。[11] B. Toghi、R. Valiente、D. Sadigh、R. Pedarsani 和 Y. P. Fallah，《体谅人类驾驶员的协同自动驾驶车辆》，载于*2021*年*IEEE/RSJ*智能机器人与系统国际会议（*IROS*）。IEEE，2021年，第4517–4524页。[12] X. Pan、B. Chen、S. Timotheou 和 S. A. Evangelou，《用于自动驾驶车辆通过交叉口的凸最优控制框架》，*arXiv*预印本 *arXiv:2203.16870*，2022年。[13] H. Xu、Y. Zhang、L. Li 和 W. Li，《利用树搜索实现无信号交叉口的协同驾驶》，*IEEE*智能交通系统汇刊，第21卷，第11期，第4563–4571页，2019年。[14] T. Kessler 和 A. Knoll，《自动驾驶的协同多车行为协调》，载于*2019*年*IEEE*智能车辆研讨会（*IV*）。IEEE，2019年，第1953–1960页。[15] K. Esterle、T. Kessler 和 A. Knoll，《自动驾驶的最优行为规划：通用混合整数形式》，载于*2020*年*IEEE*智能车辆研讨会（*IV*）。IEEE，2020年，第1914–1921页。[16] C. Burger 和 M. Lauer，《使用混合整数二次规划进行多车辆协同轨迹规划》，载于*2018*年第*21*届智能交通系统国际会议（*ITSC*）。IEEE，2018年，第602–607页。[17] S. Boyd、N. Parikh、E. Chu、B. Peleato 和 J. a. Eckstein，《通过交替方向乘子法实现分布式优化与统计学习》，《机器学习基础与趋势》，第3卷，第1期，第1–122页，2011年。[18] F. Rey、Z. Pan、A. Hauswirth 和 J. Lygeros，《用于协调与碰撞规避的完全分布式交替方向乘子法》，载于*2018*年欧洲控制会议（*ECC*）。IEEE，2018年，第825–830页。[19] Z. Wang、Y. Zheng、S. E. Li、K. You 和 K. Li，《基于交替方向乘子法的大规模网联车辆协同自动化并行最优控制》，载于*2018*年第*21*届智能交通系统国际会议（*ITSC*）。IEEE，2018年，第1633–1639页。[20] X. Zhang、Z. Cheng、J. Ma、S. Huang、F. L. Lewis 和 T. H. Lee，《基于半定松弛的交替方向乘子法：网联自动驾驶车辆的协同规划与控制》，*IEEE*智能交通系统汇刊，第23卷，9251，2021 7，9240 no. pp. – .

[21] A. D. Saravanos、Y. Aoyama、H. Zhu 和 E. A. Theodorou，《面向大规模多智能体控制的分布式微分动态规划架构》，*arXiv* 预印本 *arXiv:2207.13255*，2022。[22] G. Banjac、F. Rey、P. Goulart 和 J. Lygeros，《通过对偶共识交替方向乘子法进行去中心化资源分配》，载于*2019*年美国控制会议（*ACC*）。IEEE，2019，第2789–2794页。[23] Z. Huang、S. Shen 和 J. Ma，《通过对偶共识交替方向乘子法实现网联自动驾驶汽车协同轨迹规划的去中心化迭代线性二次调节器》，*arXiv* 预印本 *arXiv:2301.04386*，2023。[24] Y. Tassa、N. Mansard 和 E. Todorov，《控制受限微分动态规划》，载于*2014*年 *IEEE* 国际机器人与自动化会议（*ICRA*）。IEEE，2014，第1168–1175页。[25] A. Dosovitskiy、G. Ros、F. Codevilla、A. Lopez 和 V. Koltun，《CARLA：一个开放的城市驾驶模拟器》，载于机器人学习会议。PMLR，2017，第1–16页。[26] J. A. Andersson、J. Gillis、G. Horn、J. B. Rawlings 和 M. Diehl，《CasADi：用于非线性优化和最优控制的软件框架》，数学规划计算，第11卷，第1–36页，2019。