

基于采样优化与双向梯度的两阶段轨迹规划方法

徐颖磊, 王 超

北京信息科技大学计算机学院, 北京

收稿日期: 2025年4月11日; 录用日期: 2025年9月11日; 发布日期: 2025年9月24日

摘 要

轨迹规划是无人驾驶车辆安全导航的关键环节, 其目的是在动态变化的环境中实时生成无碰撞、高效的轨迹。为提高无人驾驶车辆轨迹规划的计算效率和生成轨迹的质量, 本文提出了一种基于采样空间优化粗搜索和双向梯度细优化的粗到细两阶段轨迹规划方法。该方法首先根据不同行驶工况生成对应的采样空间, 进而在生成的离散化粗全局空间中进行高效搜索, 以快速探索可能的轨迹; 进一步在连续的细化局部空间中进行双向梯度下降优化, 以精细化调整轨迹, 降低轨迹的成本。通过这种两阶段的方法, 不仅能够加快规划速度, 还能生成更优的轨迹, 有效提升了轨迹规划的性能。实验表明, 该方法对比基线运行时间减少了16.1%、轨迹总代价降低了12.6%。

关键词

轨迹规划, 仿真实验, 梯度下降, 样本空间

Two-Stage Trajectory Planning with Sampling Optimization and Bidirectional Gradient

Yinglei Xu, Chao Wang

School of Computer, Beijing Information Science & Technology University, Beijing

Received: Apr. 11th, 2025; accepted: Sep. 11th, 2025; published: Sep. 24th, 2025

Abstract

Trajectory planning is a critical component for the safe navigation of autonomous vehicles, aiming to generate collision-free and efficient trajectories in real-time within dynamically changing

文章引用: 徐颖磊, 王超. 基于采样优化与双向梯度的两阶段轨迹规划方法[J]. 人工智能与机器人研究, 2025, 14(5): 1196-1206. DOI: 10.12677/airr.2025.145113

environments. To improve the computational efficiency of trajectory planning and the quality of generated trajectories, this paper proposes a coarse-to-fine two-stage planning method based on sampling-space-optimized coarse search and bidirectional gradient fine optimization. The method first generates adaptive sampling spaces according to different driving scenarios, enabling efficient exploration of possible trajectories through discretized coarse global search. Then, it performs bidirectional gradient descent optimization in a continuous refined local space to fine-tune the trajectory and minimize its cost. By leveraging this two-stage approach, the method not only accelerates planning speed but also produces higher-quality trajectories, significantly enhancing overall planning performance. Experiments demonstrate that compared to baseline methods, this approach reduces runtime by 16.1% and decreases total trajectory cost by 12.6%.

Keywords

Trajectory Planning, Simulation Experiment, Gradient Descent, Sample Space

Copyright © 2025 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

无人驾驶车辆的规划控制系统在动态环境中承担关键决策任务, 是自动驾驶的核心模块, 涵盖轨迹规划、路径规划与行为规划。其中, 轨迹规划需高效处理复杂的高维优化问题, 对算法性能和计算能力要求极高。

现有轨迹规划方法主要分为四类: 图搜索类算法, 采样类算法, 优化类算法, 学习类算法。图搜索类算法基于离散图结构建模, 依赖先验静态地图信息, 无法有效处理动态障碍物或环境突变, 每次环境更新均需重新初始化搜索过程[1]。基于采样方法能够在规划过程中综合考虑各类条件约束, 但规划的路径存在曲折、平滑性差等问题[2]。基于深度学习的算法, 将传感器输入直接映射到控制输出, 可直接输出近似人类驾驶习惯的规划结果, 但难以泛化到极端场景, 且该方法解释性较差[3]。优化类算法将轨迹规划建模为数值优化问题, 通过目标函数和约束条件迭代求解最优轨迹, 虽能生成平滑且符合动力学约束的路径, 但对初始解敏感且计算复杂度较高, 实时性常受限于非线性约束的求解效率[4]。

其中, 基于采样的方法因计算效率高而受到广泛关注, 但其性能受限于采样分辨率, 且在高度动态环境中效果有限。在采样法中, 横向偏移采样是一种常见策略。例如, 有研究[5][6]尝试在不同横向偏移的状态空间中生成车道引导样本, 但存在轨迹偏离参考线时的平滑性和不连续性问题。另外有研究[7][8]开发了多种采样模式, 形成格状或树状结构, 以降低问题复杂性。这些方法的解质量受限于采样分辨率, 且在高度动态环境中效果有限。为提高效率, 研究[9]在生成轨迹前移除与障碍物碰撞的节点, 或采用预定义的采样模板。有研究[10]中描述的方法利用车道中心线作为参考曲线, 生成一系列与参考曲线具有不同横向偏移的轨迹样本。这些方法简单而有效, 适用于曲率较小的道路, 并且几乎不需要计算能力, 以及百度开发的方法[11]通过减少规划期间执行的必要碰撞检查次数实现了更快的性能。文献[12]注意到由于违反车辆运动学约束, 部分轨迹样本被删减, 并设计了一个用于生成符合约束的轨迹样本的车辆模型的闭环正向模拟。最近的研究 FISS [13]通过减少碰撞检查次数, 加快了规划速度。其改进的方法 FISS+[14]通过利用先验信息和粗细两阶段框架, 进一步提升了计算效率, 但其搜索阶段仍有优化空间。

为提高轨迹规划的计算效率与轨迹质量, 本研究在 FISS+框架基础上提出了一种改进的样本空间生成方法。该方法通过优化粗搜索阶段的采样策略显著提升了计算速度, 同时在细化阶段引入双向梯度下

降算法以降低求解成本。基于 CommonRoad 开环场景的多组对比实验表明, 所提算法在计算效率和轨迹质量方面均优于基线方法。

2. 轨迹规划整体框架

轨迹规划整体的框架结构如图 1 所示, 整体框架由搜索阶段和细化阶段构成, 粗阶段首先根据工况生成对应样本空间, 并估计轨迹的先验成本, 最后通过快速迭代搜索和验证得到粗解; 细阶段根据得到的粗解在周围连续空间内进行搜索和双向梯度下降优化, 并对得到的轨迹进行验证和检测得到最终的轨迹。两个阶段均采用快速迭代和采样策略。该策略旨在通过在采样空间中的高优先级区域周围生成更少但更集中的轨迹样本, 从而以高效的精细分辨率方式生成高质量的解决方案。本文改进点为图中深蓝色部分, 主要是对粗阶段中样本空间的简化和对细化阶段梯度下降优化方法的细化。

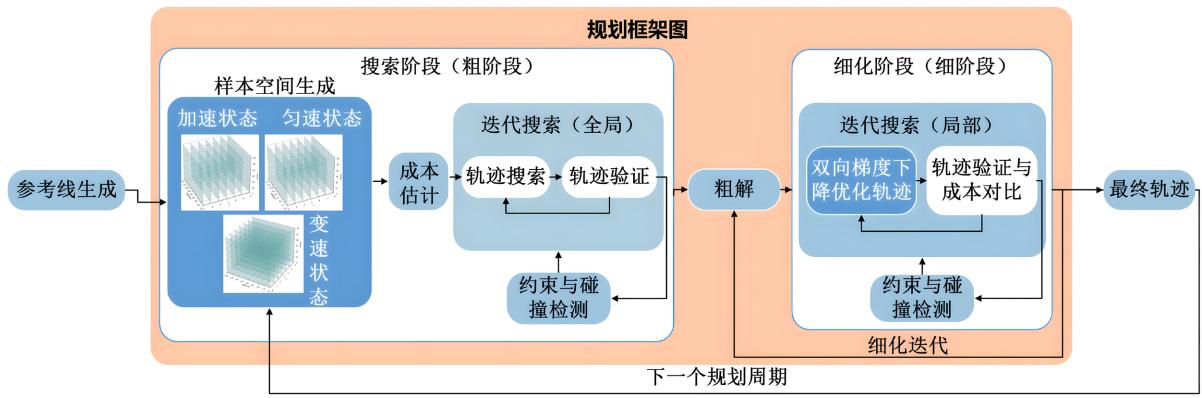


Figure 1. Trajectory planning framework diagram

图 1. 轨迹规划框架图

3. 规划算法实现

3.1. 行驶工况划分

本文采用的规划方法是基于采样的, 基于采样的方法通常根据一系列设计的成本项来检查候选轨迹。这些成本项类似于约束条件, 这些成本项将涵盖驾驶的平稳性、安全性和舒适性方面。成本项的设计通常取决于车辆类型和实际应用。但采样最终的目标是需要连接初始状态与目标区域 X_{end} 的一系列有限轨迹, 表示为: $\Pi_s = \{\pi_s(d, v, t) | d \in D, v \in V, t \in T\}$ 这里, D 、 V 和 T 分别代表横向距离、速度和时间维度上的有限采样参数集合。因此, 选择最佳轨迹的过程转化为寻找最优的采样参数组合(d 、 v 和 t)的问题:

$$\begin{aligned} \pi_s^* &= \underset{\pi_s(d, v, t)}{\operatorname{argmin}} J(\pi_s(d, v, t)), d \in D, v \in V, t \in T \\ \text{subj.to } \pi_s(t) &\in X_r(t), \forall t \in T \\ g(\pi_s(t)) &\leq 0, \forall t \in T \end{aligned} \quad (1)$$

其中 π_s^* 是初始状态 x_{init} 和目标状态 X_{end} 的最佳轨迹, 同时, 轨迹 π 必须满足安全标准 $\pi_s(t) \in X_r(t)$, 并满足由车辆运动学和交通法规等引起的各种约束 $g(\pi_s(t))$ 。执行上述操作通常需要构建三维样本空间(横向距离、速度、时间), 并在该空间进行全局搜索, 从而需要耗费大量资源。因此本文对此进行改进, 核心是通过车辆的状态分类简化三维样本空间, 同时增加粗搜索阶段搜索的时间跨度, 减少方法计算的频率, 实现计算效率的大幅提升。

根据车速将行驶状态分为三类：加速状态：车辆从静止起步至达到设定巡航速度的阶段；匀速状态：维持当前车辆巡航速度的稳态行驶阶段，通常由加速状态自动切换得到；变速状态则是应对环境突变或规则约束的速度动态调节阶段。跟车距离条件的具体设置见表 1 所示。车辆状态由自身运动参数和环境信息共同决定。

Table 1. Safe following distance for vehicles
表 1. 车辆安全跟车距离

速度区间(km/h)	推荐跟车车距(m)
>100	≥100
60~100	等于车速数值
50~60	≥50
40 及以下	≥30

这三种车速状态之间的转换关系如图 2 所示。当车速达到巡航阈值时，系统从加速状态无缝过渡至匀速状态；当感知模块实时检测到路径偏移量超限或前车距离突破安全阈值时，匀速状态即刻转入变速状态；而一旦环境约束解除，系统将重新回归稳态巡航。

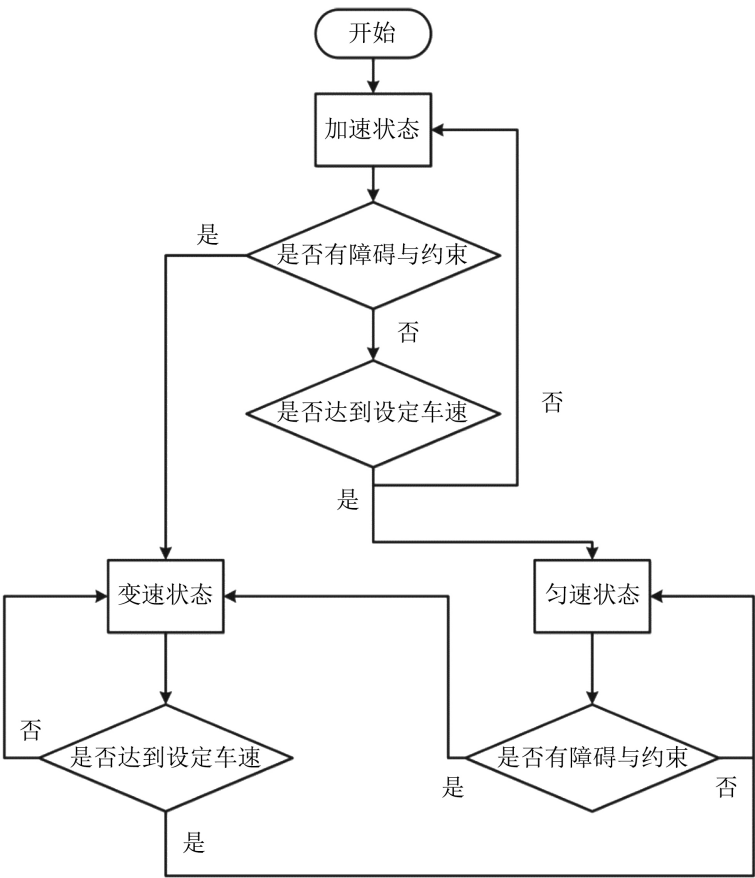


Figure 2. Diagram of vehicle condition conversion
图 2. 车辆工况转换图

3.2. 样本空间与代价函数建立

根据车辆的状态建立相关的采样空间, 在轨迹规划中, 所有轨迹样本都可以用初始状态 x_i 和唯一采样的最终状态 x_e 表示。这两个状态将定义用于插值纵向和横向轨迹分量的四次和五次多项式的形状。由于生成所有轨迹样本的效率极低, 因此本节仅获取它们的初始状态和最终状态而不是生成整个轨迹:

$$\begin{aligned} x_i &= [s_i, \dot{s}_i, \ddot{s}_i, d_i, \dot{d}_i, \ddot{d}_i, T_i]^T \\ x_e &= [s_e, \dot{s}_e, \ddot{s}_e, d_e, \dot{d}_e, \ddot{d}_e, T_e]^T \end{aligned} \quad (2)$$

其中 s 为车辆纵向坐标和 d 为车辆横向坐标, T 为未来的采样时间范围。

本文建立的样本空间如图 3 所示, 绿色平面序列沿 z 轴层级化排列, 每个平面代表一个周期计算的可行域。平面从下到上按照时间递增的顺序排列的, 通过将时间轴映射为空间层级关系, 减少了全局搜索的计算负担。图 3 中, (a) 为加速状态, 在每个周期的道路中心线采样一个速度点, 且速度值按周期次序线性递增; (b) 为匀速状态, 同样的在每个周期的道路中心线采样一个速度点, 各周期速度锁定为巡航设定值; (c) 为变速状态, 每个周期均把速度和横向距离两个维度等距离分割, 其中, 横向距离轴以车道中心为基准向两侧扩展, 轴的边界值为车道大小的二分之一, 速度轴从最低安全速度到最大限速均匀离散。

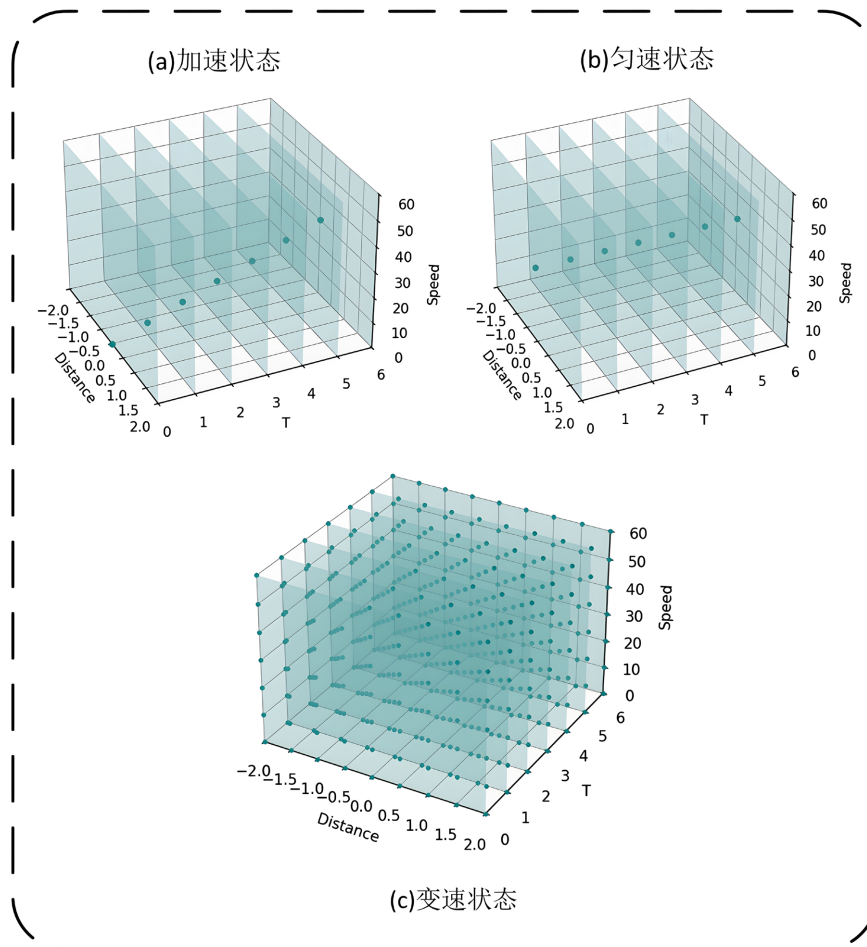


Figure 3. Schematic diagram of the sample space

图 3. 样本空间示意图

在采样空间生成后, 本节构建了相应的成本函数, 用于评估车辆从初始状态到目标状态之间的优先级, 从而筛选出当前阶段的最优解。该阶段的成本函数包括静态成本和启发式成本。其中静态成本用于预测轨迹的总体代价, 具体如下所示:

$$C_{static} = \begin{cases} C_d = \omega_d \left(\frac{d_e - d_{ref}}{d_{max}} \right)^2 \\ C_v = \omega_v \left(\frac{\dot{s}_e - v_{ref}}{v_{ref}} \right)^2 \end{cases} \quad (3)$$

静态成本的计算仅依赖于轨迹样本的初始状态和最终状态的横向偏移量和行驶速度。其中 ω 是对应每个成本项的权重大小。 d_{ref} 和 v_{ref} 是给定的与参考线的横向偏移量和参考行驶速度。

同时, 为更好地预测到达最终状态的成本, 本节建立了基于历史规划的启发式成本函数, 利用上一次计算的最优结果进行预测, 从而快速有效地估算到达最终状态的成本:

$$C_h = \omega_h \left(\frac{L(x_e, x_{e,prev}^*)}{L_{max}} \right)^2$$

$$C_{est} = C_{static} + C_h \quad (4)$$

其中, $L(x_e, x_{e,prev}^*)$ 表示本周期的状态 x_e 与上周期得到的最优结果的状态 $x_{e,prev}^*$ 在采样空间中的欧几里得距离, L_{max} 是 $x_{e,prev}^*$ 到边界的最大欧几里得距离。

在搜索阶段生成局部最优轨迹后, 需通过动态轨迹成本评估其优劣, 主要考虑: 轨迹平滑性代价: 衡量轨迹的平滑程度; 道路代价: 评估轨迹与道路边界和中心线的贴合度; 障碍物代价: 反映轨迹与周围障碍物的距离。总代价函数为三者之和, 确保生成的轨迹更适应实际环境。动态成本仅在轨迹生成后计算, 用于筛选粗解。

(1) 平滑性代价成本:

平滑性代价主要考虑的是生成轨迹曲线的本身的性质, 确保轨迹的平滑性, 提高乘坐舒适性, 并减少能量消耗。平滑性代价为:

$$C_{smooth} = \omega_1 \int f'(s)^2 ds + \omega_2 \int f''(s)^2 ds + \omega_3 \int f'''(s)^2 ds \quad (5)$$

其中, ω 是权重系数, 第一项衡量的轨迹的斜率, 第二项衡量的是轨迹的曲率, 最后一项衡量的是轨迹曲率的变化率。

(2) 道路代价成本:

道路代价成本主要受车道边界和车道中心线的影响, 车辆应该尽量行驶在车道中心线上, 同时要避免车辆行驶出道路边界外, 因此在车道中心线附近的代价最低, 车道边界附近的代价最高, 因此建立相应的道路代价成本函数为:

$$C_{road} = \omega_{road} \int \left(\frac{L}{2} - l \right)^2 \left(\frac{L}{2} + l \right)^2 \quad (6)$$

其中, ω_{road} 为道路代价权重, l 为轨迹点在 Frenet 坐标系横向方向下的坐标, L 为道路的车道宽度。

(3) 障碍物代价成本:

障碍物代价成本为轨迹上所有离散点与障碍物的代价之和, 而评价两者的障碍物成本代价值可以通过人工势场法来进行有效的评价, 一般人工势场法通过点斥力势场函数来描述障碍物的代价:

$$U(X) = \begin{cases} \frac{1}{2} k_{rep} \left[\frac{1}{\rho(X, X_0)} - \frac{1}{\rho_0} \right], & \rho(X, X_0) \leq \rho_0 \\ 0, & \rho(X, X_0) > \rho_0 \end{cases} \quad (7)$$

其中, k_{rep} 是斥力势场系数, X_0 是障碍物坐标, $\rho(X, X_0)$ 是车辆当前坐标和障碍物坐标的欧几得距离, ρ_0 是斥力势场的作用范围。

上式构建的势场函数在障碍物周围形成了圆形的势场范围, 但车辆实际运行中, 纵向的速度变化远远大于横向的速度变化, 因此应该设置为纵向方向为长轴, 横向方向为短轴的椭圆范围。此外, 需要车辆与障碍物的相对位置, 车辆后方存在障碍物, 车辆应该速度加快, 前方存在障碍物, 车辆速度应该减慢。因此本节引入了二阶高斯分布函数作为障碍物代价函数:

$$V_{obs} = \frac{C}{\sigma_1 \sigma_2 \sigma_v \sqrt{8\pi^3}} \exp \left\{ -\frac{1}{2} \left[\frac{(s_g - s_{obs})^2}{(\sigma_1 \sigma_2)^2} + \frac{(l_g - l_{obs})^2}{\sigma_v^2} \right] \right\} \quad (8)$$

其中, s 、 l 为轨迹点在 Frenet 坐标下的对应值。 σ_1 为车辆与障碍物的安全距离, σ_2 为横向车道宽度, σ_v 为纵向方向上速度的影响参数:

$$\sigma_v = B_1 \tanh[\operatorname{sgn}(s_{car} - s_{obs}) \varepsilon (v_{obs} - v_{car})] + B_2 \quad (9)$$

其中 B_1 为速度影响权重值; v_{car} , v_{obs} 为车辆和障碍物的速度大小; ε , B_2 是斥力形状影响参数。

3.3. 粗阶段搜索与细化阶段优化

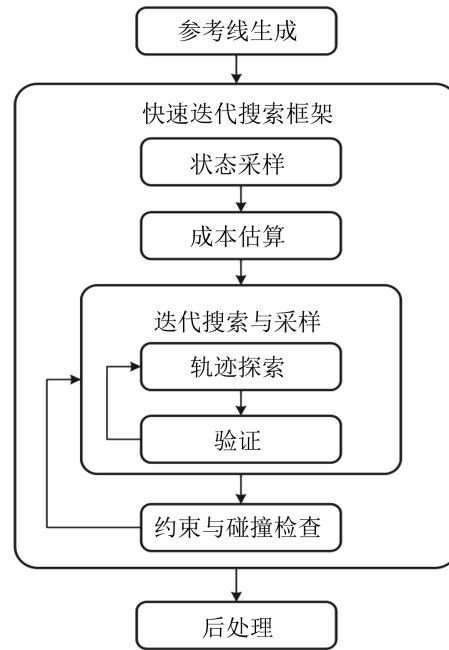


Figure 4. Quickly iterate on the search strategy flowchart
图 4. 快速迭代搜索策略流程图

生成候选轨迹通常是计算量最大的步骤。虽然任何轨迹都可能最优, 但实际情况下, 靠近参考线(如车道中心线)、接近期望速度或规划范围更长的轨迹往往成本更低, 成为最优解的概率更高。因此, 对所

有候选轨迹均等计算是低效的。为提高效率, 可预先评估轨迹的潜在成本, 优先搜索高概率候选, 排除低概率轨迹, 从而加速收敛。具体流程如图 4 所示。

在该流程中, 在 Frenet 坐标系下建立参考线之后, 将对当前周期的车辆的最终状态进行采样, 而不是把实际轨迹给采样出来。然后, 根据历史轨迹先验成本估算初始状态和最终状态的成本, 并最低成本到最高成本排序后放到优先级队列中。成本最低暂为最佳轨迹, 当目前最佳轨迹通过所有约束和碰撞检测后, 设置为粗阶段的解, 否则重新迭代搜索下一个低成本的轨迹。

计算出估计成本 C_{est} 后, 方法将采样得到的最终状态按照成本从低到高排序, 并存储在优先级队列中。粗阶段的搜索和采样模块每次从队列中取出成本最低的状态, 并以此状态为起点进行迭代搜索。该模块在离散化的粗采样空间中运行。采用贪婪搜索策略, 通过比较轨迹样本及其邻居的成本, 识别局部最优轨迹。当所有边界轨迹(即成本较低的轨迹)被探索完毕后, 算法停止并输出局部最优解。

由于粗阶段得到的粗解受限于离散空间, 为了在连续空间中得到更优的轨迹, 本节设计了细化阶段的算法。该算法以搜索阶段得到的粗解为起点, 在其邻近空间采用连续搜索策略进行优化。细化阶段可以重复执行, 以逐步优化轨迹, 进一步提升规划质量, 细化阶段搜索算法为:

算法 1 细化阶段搜索算法

```

1:    $x_{guess} \leftarrow Q_\pi$ ;  $x_{nabr} \leftarrow X_{nabr}$ ;
2:   for all  $x \in X_{nabr}$  do
3:       generate  $\pi \leftarrow f(x)$ 
4:       insert  $\pi$  into  $Q_\pi$ 
5:       compute  $C(\pi) \leftarrow C_{static} + C_h$ 
6:   end for
7:    $\frac{\delta C}{\delta dim} \leftarrow \frac{C(f(x_{guess} + \epsilon e_{dim})) - C(f(x_{guess} - \epsilon e_{dim}))}{2\epsilon}$ 
8:    $\nabla C \leftarrow \frac{\delta C}{\delta d} e_d + \frac{\delta C}{\delta v} e_v + \frac{\delta C}{\delta t} e_t$ 
9:    $x_{guess,i+1} \leftarrow x_{guess,i} - \alpha \nabla C$ 
10:  generate  $\pi_{guess} \leftarrow f(x_{guess})$ 
11:  insert  $\pi_{guess}$  into  $Q_\pi$ 

```

在算法执行过程中, 在粗解周围的连续采样空间中, 从每一维上的两个方向选取与粗状态距离较近的邻域样本, 通过公式计算当前状态的局部梯度, 并采用梯度下降的方法对粗轨迹进行优化:

$$\nabla C = \nabla C(f(x_s(d, v, t))) = \frac{\delta C}{\delta d} e_d + \frac{\delta C}{\delta v} e_v + \frac{\delta C}{\delta t} e_t \quad (10)$$

$$x_{guess,i+1} = x_{guess,i} - \alpha \nabla C$$

其中, d 、 v 、 t 是采样空间的三个维度, 横向距离, 速度, 和时间, α 是每次迭代的步长。 x 是搜索阶段生成的粗解, $f(x)$ 是用于生成轨迹的函数, C 是轨迹的成本代价。由于一般梯度下降法采用的是单侧差分来计算各个维度方向的偏导数, 可能会引入较大的误差, 因此本节采用的是中分差分来计算各个维

度的偏导数:

$$\frac{\delta C}{\delta dim} = \frac{C(f(x_{guess} + \epsilon e_{dim})) - C(f(x_{guess} - \epsilon e_{dim}))}{2\epsilon} \quad (11)$$

其中, dim 是每个维度即表示 d 、 v 、 t , e_{dim} 表示每个维度的标准基向量, ϵ 是一个很小的正数。

在细化阶段, 所有生成的轨迹样本都存储在优先级队列 Q_π 中。每次细化迭代时, 优先级队列 Q_π 中的第一个候选将被选为新的起始状态。如果细化过程因达到最大迭代次数或达到时间限制而停止, 算法将检查优先级队列 Q_π 中的轨迹候选, 直到找到可行轨迹并返回作为最终的细化轨迹 π_s^* 供车辆执行。

4. 实验

为了定量评估本算法的效率, 本文采用公共规划数据集 CommonRoad 上对其性能进行了基准测试, 同时将该算法与 FISS、FISS+和 FOP 进行了详细的对比实验。为了准确对比这些算法的运行时间, 所有仿真实验均在配备 AMD Ryzen 7 6800HS with Radeon Graphics 3.20 GHz CPU 和 16GB RAM 的设备上进行, 以确保运行时间的准确对比。

测试在 100 个选定的 CommonRoad 开环场景中进行, 每个场景包含小车道网络, 模拟车道、路口、交通信号灯及交通规则, 同时包含各种动态障碍物。所有方法均使用地面真实障碍物轨迹进行碰撞检测。一个场景的持续时间约为 10 到 15 秒, 每个规划周期按照 0.1 秒的时间步长进行划分。在每个规划周期内, 自动车辆只执行规划轨迹中的第一个状态, 然后进入下一个规划周期。一旦车辆成功到达目标区域, 将根据 CommonRoad 基准工具中定义的成本函数对自车辆的执行轨迹进行总结和评估。

为了分析算法在不同样本数量下的计算效率, 本文在所有其他参数保持不变的情况下, 增加了各维度的样本数量进行实验。各个方法的平均运行时间(以秒为单位)、每个规划周期生成的样本数以及平均成本如表 2 所示。

Table 2. Comparison results of trajectory planning algorithms
表 2. 轨迹规划算法对比结果

样本数目	方法	平均运行时间	平均代价
125	FOP+	0.239	16.100
	FISS	0.042	18.423
	FISS+	0.100	16.126
	ours	0.087	13.231
512	FOP+	0.532	14.988
	FISS	0.160	17.631
	FISS+	0.297	14.508
	ours	0.221	13.081
1000	FOP+	0.793	13.270
	FISS	0.200	16.531
	FISS+	0.413	13.320
	ours	0.380	12.471

通过比较各种方法的运行时间和平均成本，可以观察到，在相同条件下本方法得到的轨迹平均总成本为所有方法中最低的。从图 5 可知，FISS 算法计算效率最优，但其规划成本较高。相比之下，本算法能够实现更低的成本和更高的计算效率，而其他方法则需要更长时间才能达到本算法所实现的相同成本水平。该方法对比基准方法 FISS+运行时间减少了 16.1%、轨迹总代价降低了 12.6%。

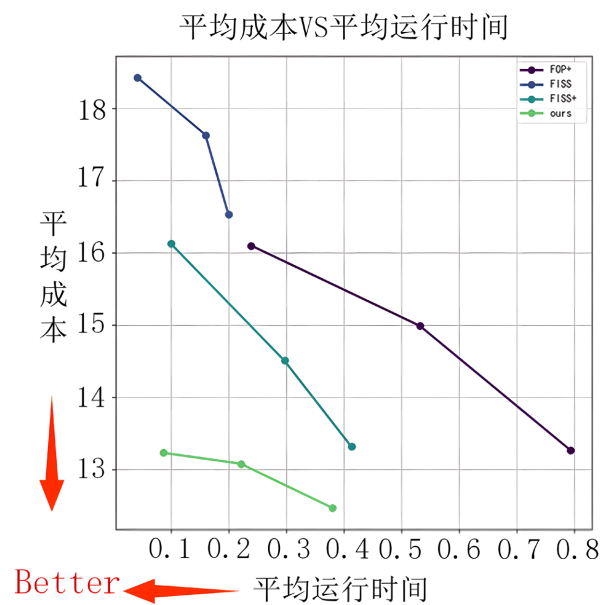


Figure 5. Trajectory average cost vs. algorithm average running time
图 5. 轨迹平均成本 vs. 算法平均运行时间

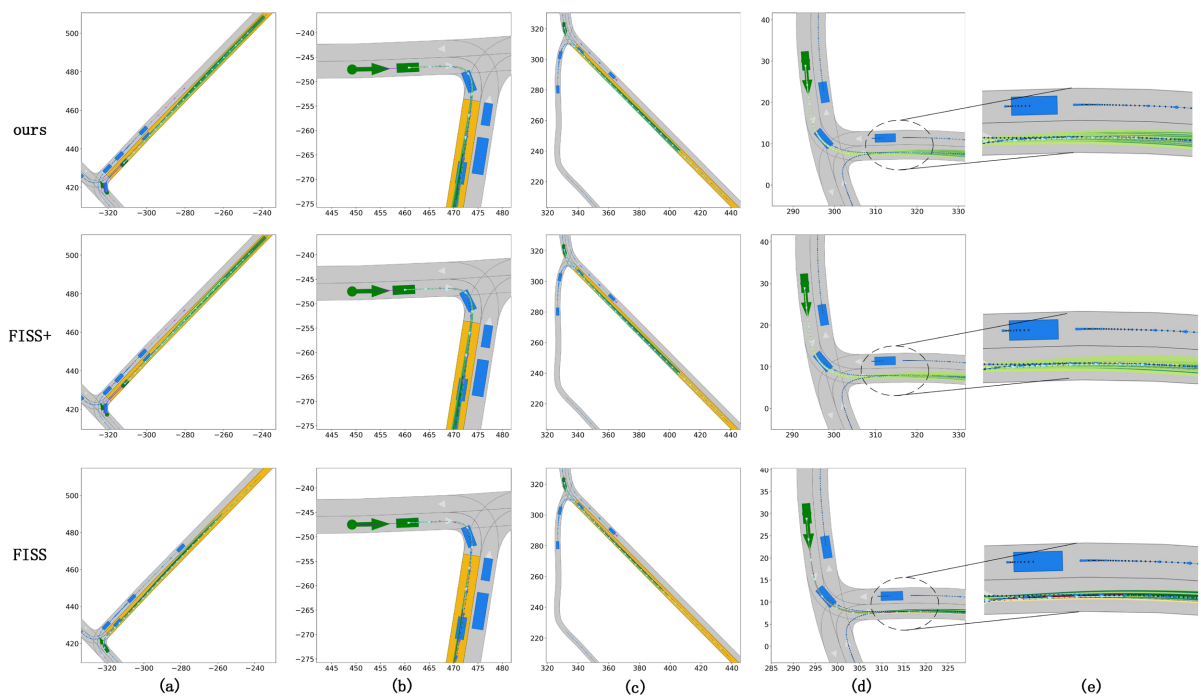


Figure 6. Comparison chart of the advantages and disadvantages of the trajectory
图 6. 轨迹优劣对比图

为了提供更直观的说明, 图 6 展示了每种算法在几个示例场景中生成的轨迹样本。在图中, 绿色车辆代表自动驾驶车辆, 蓝色车辆代表静态或动态障碍物, 黄色区域表示目标区域, 而自动驾驶车辆前方的线条则为算法生成的轨迹。为了更清晰地展示, 每张图片中样本总数选择为 125。图中的轨迹从绿色到红色表示成本逐渐增加, 通过观察每种算法生成的轨迹颜色变化判断算法的效果: 本方法生成的轨迹主要集中在低成本区域。在图中局部放大视图 6(e) 中, 可以清晰地看到 FISS 和 FISS + 算法生成的轨迹包括红色和黄色线, 代表较高成本, 而本算法生成的轨迹成本更低。

5. 结论

本文构建了一套完整的无人驾驶轨迹规划算法分层架构与优化体系, 通过轨迹规划框架构建、高效采样与代价评估、快速搜索与优化机制的优化策略, 在保证轨迹质量的同时提高了规划效率, 为无人驾驶轨迹规划提供了有效的解决方案。

参考文献

- [1] Wang, Y., Wang, C., Zhao, W. and Xu, C. (2021) Decision-Making and Planning Method for Autonomous Vehicles Based on Motivation and Risk Assessment. *IEEE Transactions on Vehicular Technology*, **70**, 107-120. <https://doi.org/10.1109/tvt.2021.3049794>
- [2] 《中国公路学报》编辑部. 中国汽车工程学术研究综述·2023[J]. 中国公路学报, 2023, 36(11): 1-192.
- [3] 孟祥福, 师光启, 张霄雁, 等. 基于深度学习的轨迹相似性度量方法研究综述[J]. 计算机科学与探索, 2025, 19(3): 623-644.
- [4] 申永鹏, 袁小芳, 赵素娜, 等. 智能网联电动汽车节能优化控制研究进展与展望[J]. 自动化学报, 2023, 49(12): 2437-2456.
- [5] Montemerlo, M., Becker, J., Bhat, S., Dahlkamp, H., Dolgov, D., Ettinger, S., et al. (2008) Junior: The Stanford Entry in the Urban Challenge. *Journal of Field Robotics*, **25**, 569-597. <https://doi.org/10.1002/rob.20258>
- [6] 韩磊, 张轮, 郭为安. 混合交通流环境下基于 MSIF-DRL 的网联自动驾驶车辆换道决策模型[J]. 北京交通大学学报, 2023, 47(5): 148-161.
- [7] Huang, Y. and Jin, C. (2023) Path Planning Based on Improved RRT Algorithm. *2023 2nd International Symposium on Control Engineering and Robotics (ISCER)*, Hangzhou, 17-19 February 2023, 136-140. <https://doi.org/10.1109/iscer58777.2023.00030>
- [8] Luo, X., Zhang, H., Zhou, X. and Yang, L. (2023) An Adaptive Path Planning Method Based on the Frenét Frame. *2023 6th International Conference on Robotics, Control and Automation Engineering (RCAE)*, Suzhou, 3-5 November 2023, 70-74. <https://doi.org/10.1109/rcac59706.2023.10398855>
- [9] 王豪杰, 马向华, 代婉玉, 等. 改进 DWA 算法的移动机器人避障研究[J]. 计算机工程与应用, 2023, 59(6): 326-332.
- [10] Taheri, E., Ferdowsi, M.H. and Danesh, M. (2018) Fuzzy Greedy RRT Path Planning Algorithm in a Complex Configuration Space. *International Journal of Control, Automation and Systems*, **16**, 3026-3035. <https://doi.org/10.1007/s12555-018-0037-6>
- [11] Hubmann, C., Schulz, J., Becker, M., Althoff, D. and Stiller, C. (2018) Automated Driving in Uncertain Environments: Planning with Interaction and Uncertain Maneuver Prediction. *IEEE Transactions on Intelligent Vehicles*, **3**, 5-17. <https://doi.org/10.1109/tiv.2017.2788208>
- [12] Ma, L., Xue, J., Kawabata, K., Zhu, J., Ma, C. and Zheng, N. (2015) Efficient Sampling-Based Motion Planning for On-Road Autonomous Driving. *IEEE Transactions on Intelligent Transportation Systems*, **16**, 1961-1976. <https://doi.org/10.1109/tits.2015.2389215>
- [13] Sun, S., Liu, Z., Yin, H. and Ang, M.H. (2022) FISS: A Trajectory Planning Framework Using Fast Iterative Search and Sampling Strategy for Autonomous Driving. *IEEE Robotics and Automation Letters*, **7**, 9985-9992. <https://doi.org/10.1109/lra.2022.3191940>
- [14] Sun, S., Chen, J., Sun, J., Yuan, C., Li, Y., Zhang, T., et al. (2023) FISS+: Efficient and Focused Trajectory Generation and Refinement Using Fast Iterative Search and Sampling Strategy. *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Detroit, 1-5 October 2023, 10527-10534. <https://doi.org/10.1109/iros55552.2023.10341498>