

自动驾驶中的决策规划算法概述

1. 引言

在一套完整的自动驾驶系统中，如果将 感知 模块比作人的眼睛和耳朵，那么决策 规划 就是自动驾驶的大脑。大脑在接收到传感器的各种 感知 信息之后，对当前环境作出分析，然后对底层控制模块下达指令，这一过程就是决策 规划 模块的主要任务。同时，决策 规划 模块可以处理多么复杂的场景，也是衡量和评价自动驾驶能力最核心的指标之一[1]。

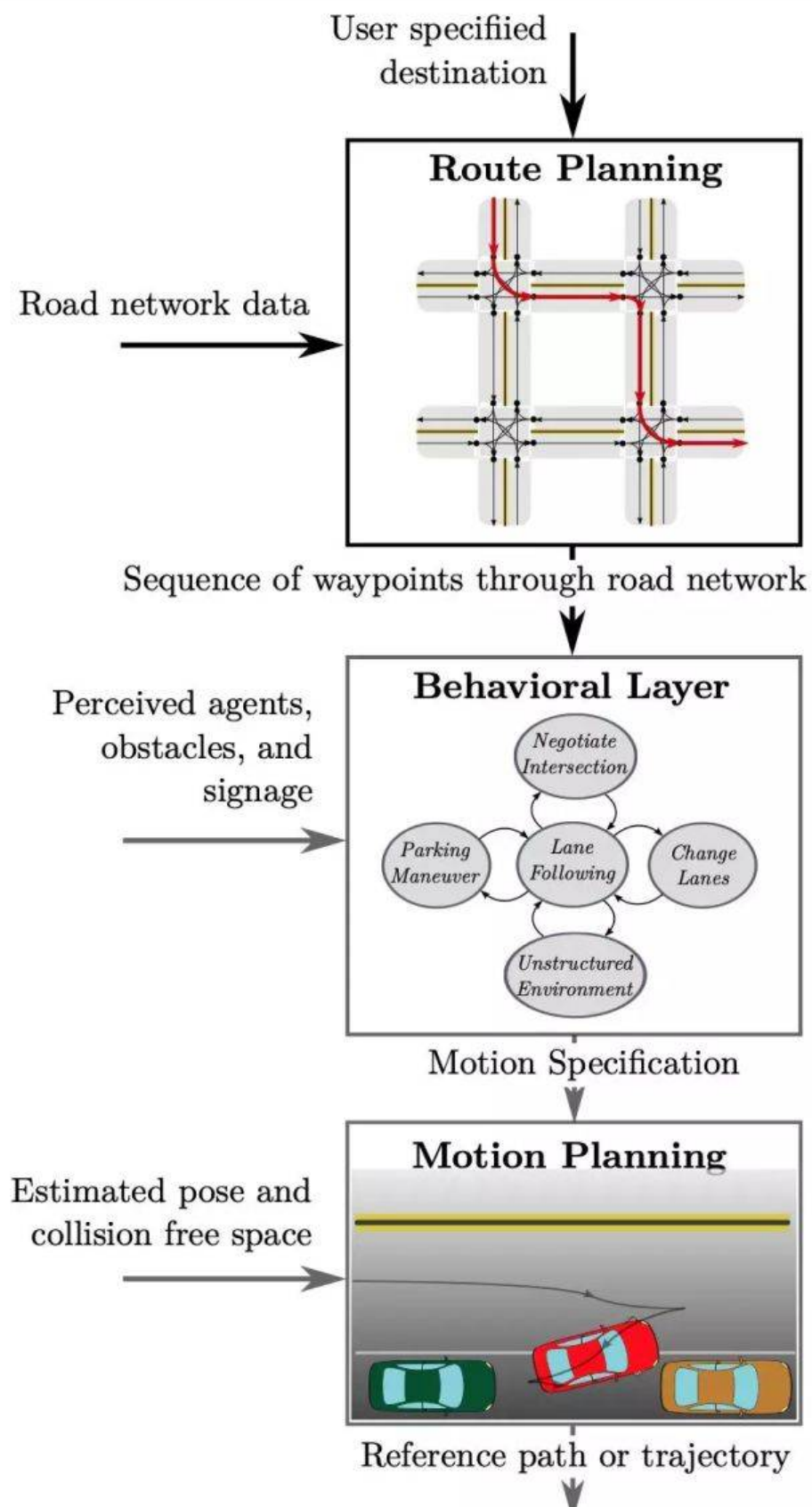


图1. 自动驾驶系统中的决策 规划 模块分层结构，引用自[2]

如图1所示，典型的决策 规划 模块可以分为三个层次。

其中，全局 路径规划（Route Planning）在接收到一个给定的行驶目的地之后，结合地图信息，生成一条全局的路径，作为为后续具体 路径规划 的参考；

行为决策层（Behavioral Layer）在接收到全局路径后，结合从 感知 模块得到的环境信息（包括其他车辆与行人，障碍物，以及道路上的交通规则信息），作出具体的行为决策（例如选择变道超车还是跟随）；

最后，运动规划（Motion Planning）层根据具体的行为决策，规划 生成一条满足特定约束条件（例如车辆本身的动力学约束、避免碰撞、乘客舒适性等）的轨迹，该轨迹作为控制模块的输入决定车辆最终行驶路径。

本文将分别介绍各层的主要作用与常见算法，并且比较各种算法的优劣性及适用情景。

2. 全局路径规划（Route Planning）

全局 路径规划 是指在给定车辆当前位置与终点目标后，通过搜索选择一条最优的路径，这里的“最优”包括路径最短，或者到达时间最快等条件。这一过程类似于我们生活中经常用到的“导航”功能，区别在于自动驾驶中使用的高精地图与我们常见的地图不太一样，在高精地图中包含了每条车道在内的更多信息。常见的全局 路径规划 算法包括Dijkstra和A算法，以及在这两种算法基础上的多种改进。Dijkstra算法[3]和A*算法[4]也是在许多 规划 问题中应用最为广泛的两种搜索算法。



图2. 全局 路径规划 示意

1. Dijkstra算法

Dijkstra算法是由计算机科学家Edsger W. Dijkstra在1956年提出，用来寻找图形中节点之间的最短路径。在Dijkstra算法中，需要计算每一个节点距离起点的总移动代价。同时，还需要一个优先队列结构。对于所有待遍历的节点，放入优先队列中会按照代价进行排序。在算法运行的过程中，每次都从优先队列中选出代价最小的作为下一个遍历的节点。直到到达终点为止。

Dijkstra算法的优点是：给出的路径是最优的；缺点是计算 时间复杂度 比较高（ $O(N^2)$ ），因为是向周围进行探索，没有明确的方向。

2. A*算法

为了解决Dijkstra算法的搜索效率问题，1968年，A算法由Stanford研究院的Peter Hart, Nils Nilsson以及Bertram Raphael发表，其主要改进是借助一个启发函数来引导搜索的过程。具体来说，A算法通过下面这个函数来计算每个节点的优先级：

$$f(n) = g(n) + h(n)$$

其中：

- $f(n)$ 是节点 n 的综合优先级。当我们选择下一个要遍历的节点时，我们总会选取综合优先级最高（值最小）的节点。
- $g(n)$ 是节点 n 距离起点的代价。
- $h(n)$ 是节点 n 距离终点的预计代价，这也就是A*算法的启发函数。

3. 行为决策（Behavioral Layer）

在确定全局路径之后，自动驾驶车辆需要根据具体的道路状况、交通规则、其他车辆与行人等情况作出合适的行为决策。

这一过程面临三个主要问题：

首先，真实的驾驶场景千变万化，如何覆盖？

其次，真实的驾驶场景是一个多智能体决策环境，包括主车在内的每一个参与者所做出的行为，都会对环境中的其他参与者带来影响，因此我们需要对环境其他参与者的行为进行预测；

最后，自动驾驶车辆对于环境信息不可能做到100%的 感知，例如存在许多被障碍物遮挡的可能危险情形。

综合以上几点，在自动驾驶行为决策层，我们需要解决的是在多智能体决策的复杂环境中，存在 感知不确定性情况的 规划 问题。可以说这一难题是真正实现L4、L5级别 自动驾驶技术 的核心瓶颈之一，近年来随着 深度强化学习 等领域的快速发展，为解决这一问题带来了新的思路和曙光。

以下将行为决策层的模型分为四类分别介绍[5]：

1. 有限状态机 模型

自动驾驶车辆最开始的决策模型为 有限状态机模型[6]，车辆根据当前环境选择合适的驾驶行为，如停车、换道、超车、避让、缓慢行驶等模式，状态机模型通过构建有限的有向连通图来描述不同的驾驶状态以及状态之间的转移关系，从而根据驾驶状态的迁移反应式地生成驾驶动作。

有限状态机模型因为简单、易行，是无人驾驶领域目前最广泛的行为决策模型，但该类模型忽略了环境的动态性和不确定性，此外，当驾驶场景特征较多时，状态的划分和管理比较繁琐，多适用于简单场景下，很难胜任具有丰富结构化特征的城区道路环境下的行为决策任务。

2. 决策树模型

决策/行为树模型[7]和状态机模型类似，也是通过当前驾驶状态的属性值反应式地选择不同的驾驶动作，但不同的是该类模型将驾驶状态和控制 逻辑 固化到了树形结构中，通过自顶向下的“轮询”机制进行驾驶 策略搜索。这类决策模型具备可视化的控制 逻辑，并且控制节点可复用，但需要针对每个驾驶场景离线定义决策网路，当状态空间、行为空间较大时，控制 逻辑 将比较复杂。另外，该类模型同样无法考虑交通环境中存在的不确定性因素。

3. 基于知识的推理决策模型

基于知识的推理决策模型由“场景特征 - 驾驶动作”的 映射 关系来模仿人类驾驶员的行为决策过程，该类模型将驾驶知识存储在 知识库 或者 神经网络 中，这里的驾驶知识主要表现为规则、案例或场景特征到驾驶动作的 映射 关系。进而，通过“查询”机制从 知识库 或者训练过的网络结构中推理出驾驶动作。

该类模型主要包括：基于规则的推理系统[8]、基于案例的推理系统[9]和基于 神经网络的映射模型[10]。

该类模型对先验驾驶知识、训练数据的依赖性较大，需要对驾驶知识进行精心整理、管理和更新，虽然基于 神经网络的映射模型可以省去数据标注和知识整合的过程，但是仍然存在以下缺点：

- 其“数据”驱动机制使得其对训练数据的依赖性较大，训练数据需要足够充分[11]；
- 将 映射 关系固化到网络结构中，其解释性较差；

- 存在“黑箱”问题，透明性差，对于实际系统中出现的问题可追溯性较差，很难发现问题的根本原因。

4. 基于价值的决策模型

根据最大效用理论，基于效用 / 价值的决策模型的基本思想是依据选择准则在多个备选方案中选择出最优的驾驶策略 / 动作[12]。

为了评估每个驾驶动作的好坏程度，该类模型定义了效用（utility）或价值（value）函数，根据某些准则属性定量地评估驾驶策略符合驾驶任务目标的程度，对于无人驾驶任务而言，这些准则属性可以是安全性、舒适度、行车效率等，效用和价值可以由其中单个属性决定也可以是由多个属性决定。

澳大利亚格里菲斯大学的Furda和Vlacic提出了多准则决策方法从候选动作集中选择最优的驾驶动作[13]；新加坡国立大学的Bandyopadhyay等人提出了基于POMDP的行为决策模型[14]，用以解决存在 感知不确定性的情况；卡内基梅隆大学的Wei J等人提出基于PCB（Prediction and-Cost-function Based）的行为决策模型[15]，其侧重点在于如何构建恰当的代价函数来指导对环境的预测；为了解决在多智能体参与的复杂环境中的决策问题，许多基于 博弈论的模型也被研究者用来推理车辆之间的交互行为[16]，[17]；此外，因为在特征提取方面的优势，深度强化学习技术也开始被广泛应用，以完成最优驾驶动作的生成[18]。

4. 运动规划（Motion Planning）

在确定具体的驾驶行为之后，我们需要做的是将“行为”转化成一条更加具体的行驶“轨迹”，从而能够最终生成对车辆的一系列具体控制信号，实现车辆按照 规划目标的行驶。这一过程称为 运动规划（Motion Planning），运动规划的概念在机器人领域已经有较长时间的研究历史，我们可以从数学的角度将它看做如下的一个优化问题：

路径规划（Path Planning）

Problem IV.1 (Optimal path planning). Given a 5-tuple $(\mathcal{X}_{\text{free}}, \mathbf{x}_{\text{init}}, X_{\text{goal}}, D, J)$ find $\sigma^* =$

$$\begin{aligned} & \arg \min_{\sigma \in \Sigma(\mathcal{X})} J(\sigma) \\ \text{subj. to } & \sigma(0) = \mathbf{x}_{\text{init}} \text{ and } \sigma(1) \in X_{\text{goal}} \\ & \sigma(\alpha) \in \mathcal{X}_{\text{free}} \quad \forall \alpha \in [0, 1] \\ & D(\sigma(\alpha), \sigma'(\alpha), \sigma''(\alpha), \dots) \quad \forall \alpha \in [0, 1]. \end{aligned}$$

图3. 路径规划 的定义

在以机器人为代表的许多场景中，我们可以认为周围的环境是确定的。在这种情况下，所谓的路径规划，是指在给定的一个状态空间 X ，寻找一个满足一定约束条件的映射 $\sigma: [0,1] \rightarrow X$ ，这些约束包括：

- 确定的起始状态以及目标点所在的区域
- 避免碰撞
- 对路径的微分约束（例如在实际问题中路径曲率不能太小，对应于其二阶导数的约束）

该优化问题的目标泛函定义为 $J(\sigma)$ ，其具体意义可以表示为路径长度、控制复杂度等衡量标准。

然而在自动驾驶问题中，车辆周围的环境是持续动态变化的，因此单纯的路径规划不能给出在行驶过程中一直有效的解，因此我们需要增加一个维度——时间 T ，相应的规划问题通常被称为轨迹规划。

轨迹规划 (Trajectory Planning)

Problem IV.2 (Optimal trajectory planning). Given a 6-tuple $(\mathcal{X}_{\text{free}}, \mathbf{x}_{\text{init}}, X_{\text{goal}}, D, J, T)$ find $\pi^* =$

$$\begin{aligned} & \arg \min_{\pi \in \Pi(\mathcal{X}, T)} J(\pi) \\ & \text{subj. to} \quad \pi(0) = \mathbf{x}_{\text{init}} \text{ and } \pi(T) \in X_{\text{goal}} \\ & \quad \pi(t) \in \mathcal{X}_{\text{free}} \quad \forall t \in [0, T] \\ & \quad D(\pi(t), \pi'(t), \pi''(t), \dots) \quad \forall t \in [0, T]. \end{aligned}$$

图4. 轨迹规划的定义

时间维度的增加为 规划 问题带来了巨大的挑战。例如，对于一个在2D环境中移动一个抽象为单点的机器人，环境中的障碍物近似为多边形的问题。路径规划 问题可以在多项式时间内求解，而加入时间维度的轨迹规划 问题已经被证明是NP-hard问题。

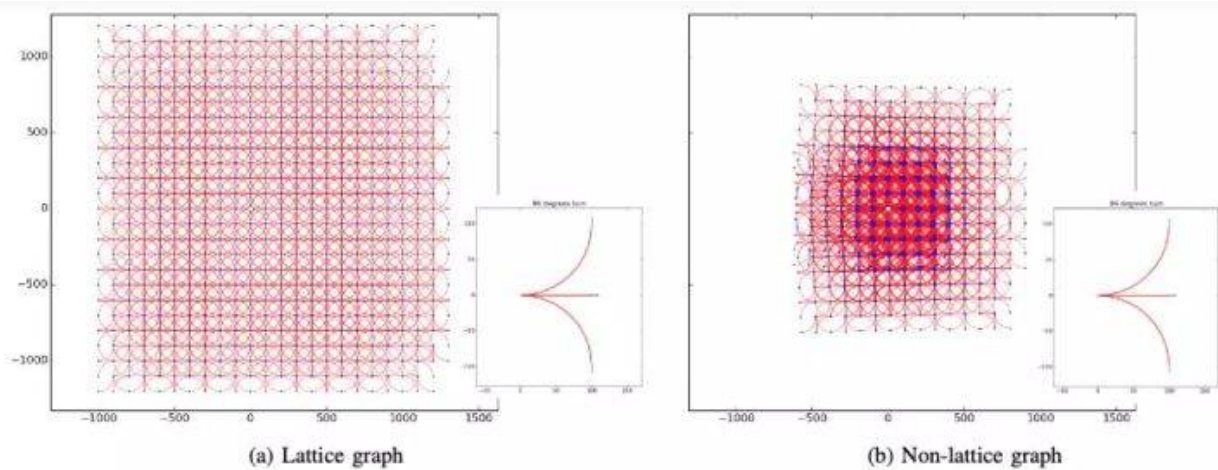
在自动驾驶的实际场景中，无论是对车辆本身还是对周围环境，建立更为精确的模型意味着对优化问题更为复杂的约束，同时也意味着求解的更加困难。因此实际采用的算法都是建立在对实际场景的近似前提下，在模型精确性和求解效率二者之间寻求一个最佳的平衡点。

下文对自动驾驶领域目前常见的几类 运动规划 算法分别进行介绍，在实际中，往往是其中几类思想的结合才能最终达到比较好的 规划 结果，并满足更多的不同场景。

1. 基于搜索的规划算法

通过搜索来解决 运动规划 问题是最朴素的思路之一,其基本思想是将状态空间通过确定的方式离散成一个图,然后利用各种 启发式搜索 算法搜索可行解甚至是最优解。

在将状态空间离散化的过程中，需要注意的是确保最终形成的栅格具有最大的覆盖面积，同时不会重复。如图5所示，左边的栅格是由直行、左转90°、右转90°这三种行为生成；而如果选择直行、左转89°、右转89°三种行为，最后就无法生成一个覆盖全部区域的栅格结构。



在将状态空间栅格化之后，我们就可以使用前文已经介绍的Dijkstra、A*搜索算法，完成最终的规划。然而在实际复杂环境中，栅格数目众多，并且环境随时间动态变化，会导致搜索结点过多，因此发展出了多种改进算法，用以处理不同的具体场景：

1) Hybrid A* 算法，在A*算法的基础上考虑了车的最大转向问题，例如限定计算的路径上车最大转向不超过 5° 。该算法目前的应用场景有车掉头（Stanford 参加DARPA 挑战赛使用的Junior车采用了该算法进行uturn），泊车等等对方向盘控制要求较高的场景。

2) D*、D*Lite算法，事先由终点向起点进行搜索，使用Dijkstra算法，存储路网中目标点到每个点的最短路径长度 k ，和该节点到目标点的实际长度值 h ，开始情况下 $k=h$ ，并且存储每个节点的上一个节点，保证能够沿着链接走下去。

计算结束后，获取了一条当时最优路径。当车行驶到某个节点时，通过传感器发现该节点已经无法通行（有障碍物），则对已存储的路网信息一些相关点的 h 值进行修改（变大），选择一个邻居点满足仍然 $h=k$ 的，即仍然是最优路径上的点，作为下一个点。

然后走到终点。该类算法适用于在未知环境中的导航以及 路径 规划，广泛用于目前各种 移动机器人 和自主车辆载具，例如“机遇号”和“勇气号”火星车。

2. 基于采样的 规划 算法

通过对连续的状态空间进行采样，从而将原问题近似成一个离散序列的优化问题，这一思路也是在计算机科学中应用最为广泛的算法。在 运动 规划 问题中，基于采样的基本算法包括概率路线图(PRM)和快速搜索随机树(RRT)算法。

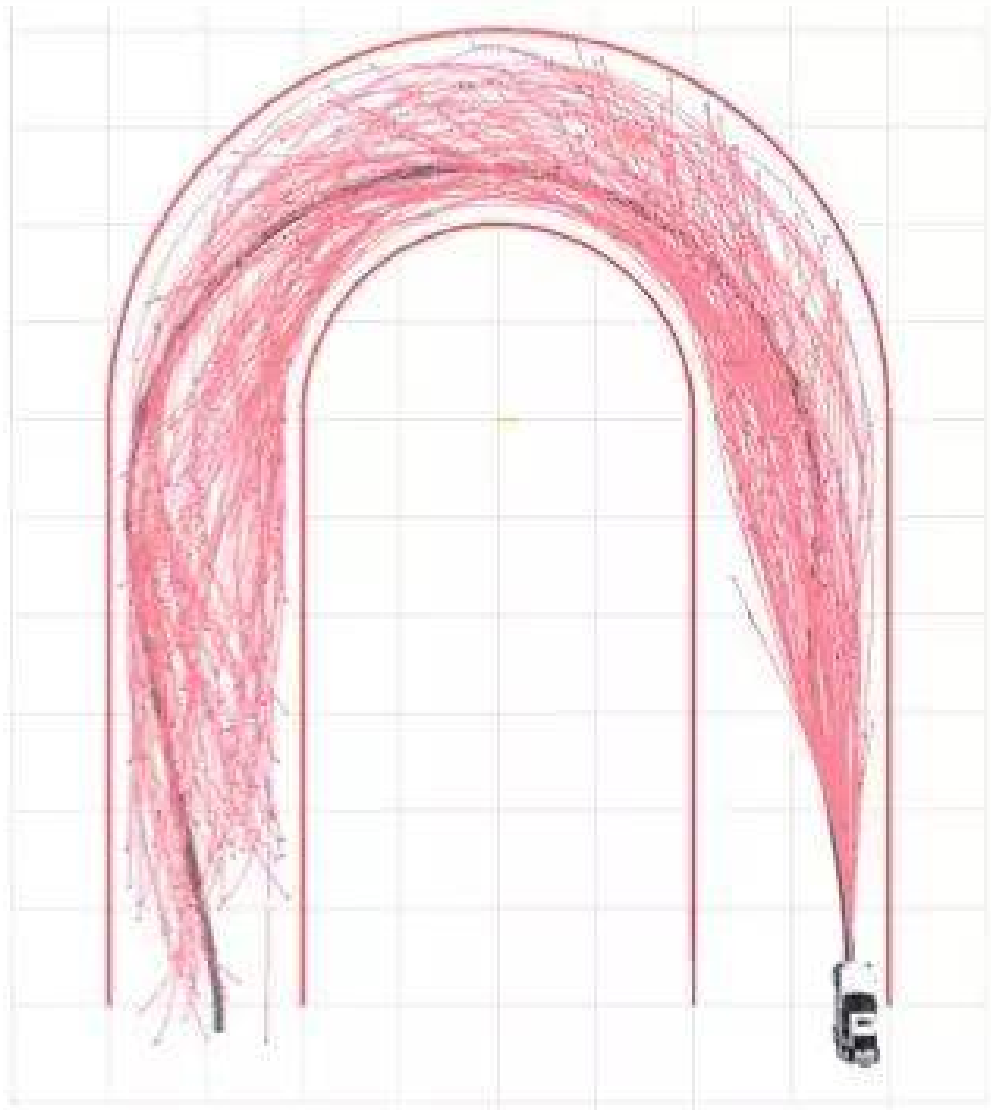


图6. 利用RRT算法实现u形弯的轨迹规划，引用自[19]

1) 基本算法：概率路线图(PRM)

- 预处理阶段：对状态空间内的安全区域均匀随机采样 n 个点，每个采样点分别与一定距离内的邻近采样点连接，并丢弃掉与障碍物发生碰撞的轨迹,最终得到一个连通图。
- 查询阶段：对于给定的一对初始和目标状态,分别将其连接到已经构建的图中,再使用搜索算法寻找满足要求的轨迹。

容易看出,一旦构建一个PRM之后,可以用于解决不同初始、目标状态的 运动规划 问题,但是这个特性对于自动驾驶 运动规划 而言是不必要的。另外PRM要求对状态之间作精确连接,这对于存在复杂微分约束的 运动规划 问题是十分困难的。

2) 基本算法：快速搜索随机树(RRT)

- 树的初始化：初始化树的结点集和边集,结点集只包含初始状态,边集为空。
- 树的生长：对状态空间随机采样，当采样点落在状态空间安全区域时，选择当前树中离采样点最近的结点，将其向采样点扩展(或连接)。若生成的轨迹不与障碍物发生碰撞，则将该轨迹加入树的边集，该轨迹的终点加入到树的结点集。

RRT是一种增量式采样的搜索方法，无须设置任何分辨率 参数。在极限情况，该搜索树将稠密的布满整个空间，此时搜索树由很多较短曲线或路经构成，以实现充满整个空间的目的。

3) 多种改进算法

从以上基础算法的描述我们可以了解到，对状态空间进行采样，可以保证得到连接起始点与终点的可行解，但由于采样过程是对整个空间进行均匀采样，因此效率很低；在复杂场景下无法实现实时求解；此外，最终 规划 结果无法保证得到的可行解是最优解。针对这些劣势，多种改进算法被提出并应用于自动驾驶问题：

- 效率改进--不均匀采样
 - RRT-Connect：同时构建两棵分别起始于初始状态和目标状态的树,当两棵树生长到一起时则找到可行解。
 - 启发式（hRRT）：使用启发式函数增加扩展代价低的结点被采样的概率。
 - 结合驾驶员模型：结合驾驶员视觉注意力模型进行偏向性采样，利用视觉特征信息引导 运动规划，使 规划 出的轨迹更符合人类驾驶行为。
 - 构造新度量RG-RRT(reachability guided RT)：常规类欧式距离的度量并不能真实反映构形或状态之间的远近，RG-RRT计算树中结点的能达集，当采样点到结点的距离大于采样点到该结点能达集的距离时,该节点才有可能被选中进行扩展。

- 加入障碍物惩罚 (RC-RRT、EG-RRT、ADD-RRT等) : 降低靠近障碍物的结点获得扩展的概率。

- 实时性改进

- anytime RRT先快速构建一个RRT,获得一个可行解并记录其代价.之后算法会继续采样,但仅将有利于降低可行解代价的结点插入树中,从而逐渐获得较优的可行解。

- Replanning将整个 规划 任务分解为若干等时间的子任务序列, 在执行当前任务的同时 规划 下一个任务。

- 最优性改进

- PRM*、RRG、RRT* : 根据随机几何图理论 (在状态空间中随机采样m个点,并将距离小于 $r(n)$ 的点连起来,就构成了随机几何图) 对标准PRM 和RRT做出改进,得到了具有渐近最优性质的PRM*、RRG和RRT*算法

3. 直接优化方法

在绝大多数情况下, 不考虑高度的变化, 自动驾驶的轨迹 规划 问题是一个三维约束优化问题 (2D空间+时间T), 因此, 我们可以采用解耦的策略, 将原始问题分解为几个低维问题, 从而大大降低求解难度。

1) Frenet坐标系

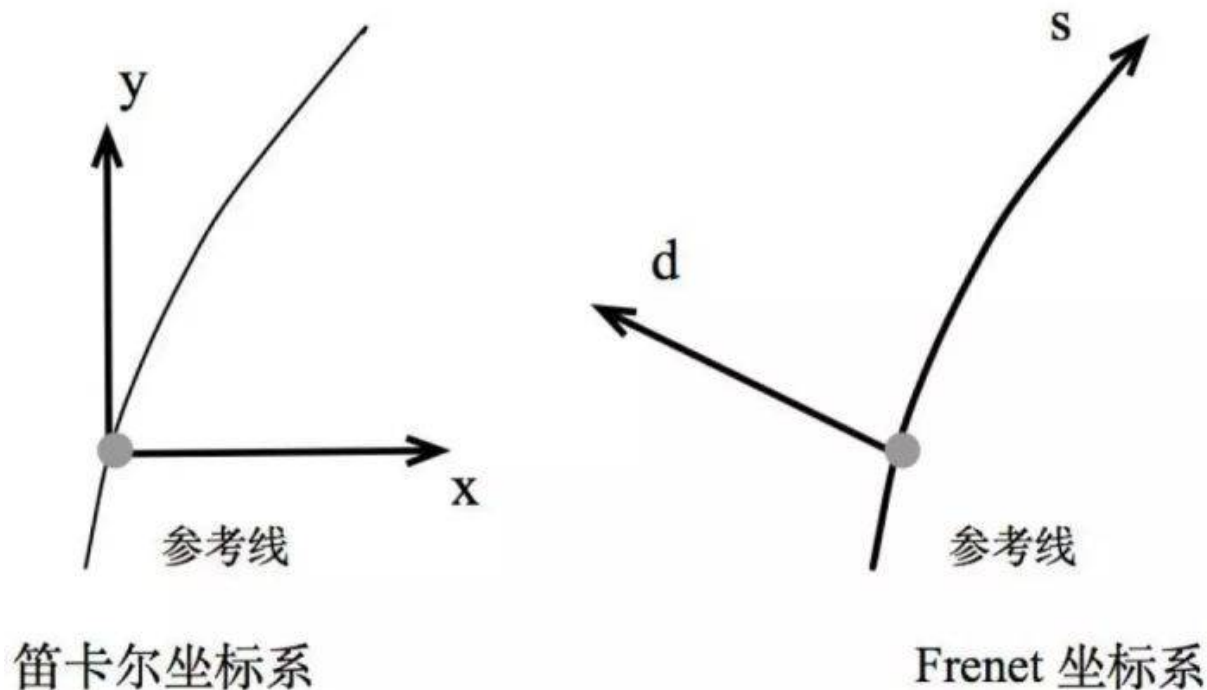


图7. Frenet坐标系

由于真实世界中的道路都是弯曲的，为了简化求解优化问题的 参数 表达，在自动驾驶中通常采用Frenet坐标系。

在Frenet坐标系中，我们使用道路的中心线作为参考线，使用参考线的切线向量 t 和法线向量 n 建立一个坐标系，如右图所示，它以车辆自身为原点，坐标轴相互垂直，分为 s 方向（即沿着参考线的方向，通常被称为纵向，Longitudinal）和 d 方向（或 L 方向，即参考线当前的法向，被称为横向，Lateral），相比于笛卡尔坐标系（左图），Frenet坐标系明显地简化了问题。

因为在公路行驶中，我们总是能够简单的找到道路的参考线（即道路的中心线），那么基于参考线的位置的表示就可以简单的使用纵向距离 S （即沿着道路方向的距离）和横向距离 L （即偏离参考线的距离）来描述。

3) 路径-速度解耦法：

在Frenet坐标系中，路径-速度解耦法分别优化路径和速度，路径优化主要考虑静态障碍物，通过动态 规划 生成一条静态的参考路径（SL维度），接着基于生成的路径，考虑对速度的 规划（ST维度）。这一过程可以不断迭代，从而实现对轨迹的实时更新。在 百度 的开源自

自动驾驶平台Apollo中采用的EM planner就是基于类似的解决思路。这一方案具有较强的灵活性，可以普适性的适用于许多场景。

除此之外，也可以选择不同的解耦方式，例如分别对纵向轨迹（ST维度）和横向轨迹（LT维度）进行规划。但需要指出的是，通过解耦的方法得到的解可能不是最优的，并且这种算法不具备完备性，在一些复杂环境可能找不到可行解。

4. 参数化曲线构造法

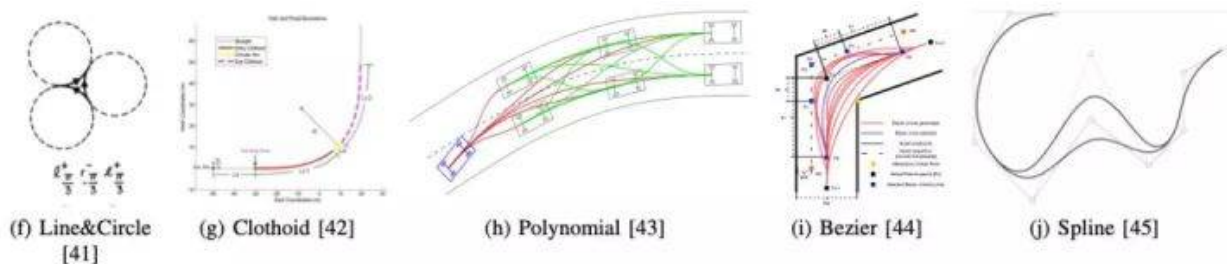


图8. 常见的参数化曲线构造法，引用自[19]

参数化曲线构造法的出发点是车辆本身的约束，包括运动学与动力学的约束，因此一般规划的路径需要是曲率连续的。这类方法根据起始点和目标点，考虑障碍物，通过构造一族符合车辆约束的曲线给出一条平滑路径。

如图8所示，常见的曲线有Dubins曲线（由直线和圆弧构成，是一种简单车辆模型Dubin模型在二维空间中的最短曲线族），回旋曲线，多项式曲线，贝塞尔曲线，样条曲线等。单纯应用参数化曲线构造法很难满足实际复杂场景，因此现在越来越多的自动驾驶系统将其与其他方法结合，用来对已经规划生成的轨迹做平滑处理，以满足车辆运动学与动力学约束。

5. 人工势场法

人工势场法是受物理学中电磁场的启发，假设障碍物和目标位置分别产生斥力和引力，从而可以沿着势场的最速梯度下降去规划路径。这类方法的一个关键问题是如何选择合适的势场函数，例如：Stephen Waydo使用流函数进行平滑路径的规划[20]，Robert Daily在高速车辆上提出谐波势场路径规划方法[21]。在简单场景下，人工势场法具有较高的求解效率，但

其存在的最大问题是可能陷入局部最小值，在这种情况下，所获得的路径不是最优，甚至可能找不到路径。

5. 算法复杂度 (Complexity)

在 规划 问题中，对于一个算法的评价除了要考虑其时间和空间复杂度之外，还要考虑其是否具有完备性和最优性，退一步的情况下考虑其是否具有概率完备性和渐进最优性。只要在了解这些性质的基础上，我们才能针对不同的实际场景设计和应用不同的算法，从而达到模型复杂和效率最优的最佳衡点。

1) 完备性 (Completeness)：如果在起始点和目标点间有路径解存在，那么一定可以得到解，如果得不到解那么一定说明没有解存在；

2) 概率完备性 (Probabilistically Completeness)：如果在起始点和目标点间有路径解存在，只要 规划 或搜索的时间足够长，就一定能确保找到一条路径解；

3) 最优性 (Optimality)：规划 得到的路径在某个评价指标上是最优的（评价指标一般为路径的长度）

4) 渐进最优性 (Asymptotically optimality)：经过有限次 规划 迭代后得到的路径是接近最优的次优路径，且每次迭代后都与最优路径更加接近，是一个逐渐 收敛 的过程

算法类别	完备性	最优性
基于搜索	完备	最优
基于采样	概率完备	渐进最优
直接优化	不完备	局部最优
参数化曲线构造	不完备	局部最优
人工势场法	不完备	局部最优

表1 常见算法比较

6. 未来发展趋势

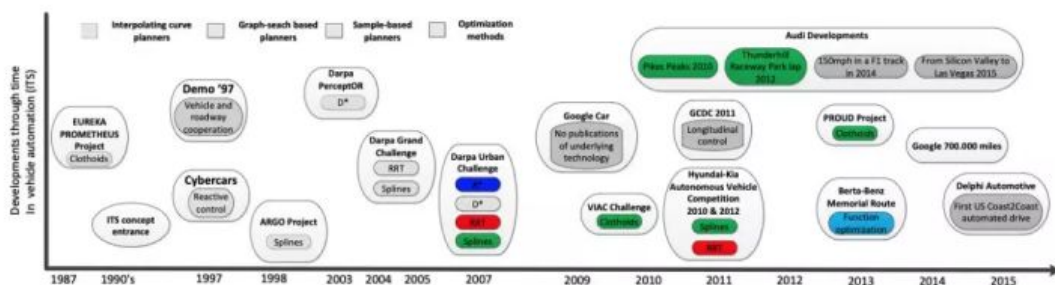


图9. 自动驾驶发展时间线及过程中重要的运动规划算法，引用自[19]

人类对自动驾驶的兴趣最早可以追溯到1925年，近年来对自动驾驶的研究热潮始于美国国防先进研究项目局（Defense Advanced Research Projects Agency, DARPA）在2004-2007年举办的3届自动驾驶挑战赛[22]，如图9所示。在此之后，上文提到的各类决策规划方法的有效性都被实际验证。同时，将运动规划方法与控制理论、状态参数估计、机器学习等多领域方法相结合的解决方案也不断出现，成为未来的发展趋势：

1) 与车辆动力学结合：将动力学 参数 评价指标和最优 规划 等结合，从最优控制角度进行 规划 是近年采用较多的方法，在这个过程中可以充分考虑车辆动力学因素，规划 出的轨迹更加合理。例如采用模型预测 控制理论（Model Predictive Control）。其不足在于：对车辆的约束越多，优化其轨迹的难度越大，较难实现在线的实时计算。

2) 与状态 参数 估计结合：状态 参数 估计可以更加准确获得车辆 参数，因此可以将状态估计器加入 规划 模块中，通过在线估计车辆状态并将其反馈给 规划 器，提高轨迹质量。例如：不同地面类型会引起车辆滑移特性的变化，进而影响车辆状态，通过结合估计 参数 实时重新 规划 轨迹，闭环 规划 从而提高轨迹安全性。

3) 与 机器学习 结合：随着以 神经网络 为代表的人工智能的快速发展，许多传统的 规划 问题也带来了新的解决思路。在自动驾驶领域，其发展趋势包括：

- 端到端模型：使用一个深度 神经网络，直接根据车辆状态和外部环境信息得出车辆的控制信号。尽管目前的端到端模型存在类似“黑箱”的不可解释性，但相信随着人类对深度 神经网络 理解的不断加深，这一方法因其突出的简洁高效优势而具有很强的发展潜力。
- 决策与 运动 规划 模块融合

自动驾驶车辆在复杂环境中作出最优决策，这一问题与强化学习的定义非常吻合，因此如前文所述，随着 深度强化学习 技术的快速发展，越来越多的研究团队开始将其应用于自动驾驶决策 规划 中，将行为决策与 运动 规划 模块相融合，直接学习得到行驶轨迹。为了解决环境奖励函数不易获得的问题，人们还提出了首先利用逆强化学习（IRL）根据人类专家演示学习，然后使用强化学习来学习最优策略。

参考文献

[1] Schwarting W, Alonso-Mora J, Rus D. Planning and Decision-Making for Autonomous Vehicles[J]. Annual Review of Control, Robotics, and Autonomous Systems, 2018, 1(1): 187–210.

[2] Paden B, Cap M, Yong S Z, et al. A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles[J]. IEEE Transactions on Intelligent Vehicles, 2016, 1(1): 33–

- [3] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, pp. 269–271, 1959.
- [4] N. J. Nilsson, "A mobile automaton: An application of artificial intelligence techniques," tech. rep., DTIC Document, 1969.
- [5] 耿新力. 城区不确定环境下无人驾驶车辆行为决策方法研究[D]. 中国科学技术大学, 2017.
- [6] Montemerlo M, Becker J, Bhat S, et al. Junior: The Stanford entry in the Urban Challenge[J]. *Journal of Field Robotics*, 2008, 25(9): 569–597.
- [7] Olsson M. Behavior Trees for decision-making in Autonomous Driving[M]. 2016.
- [8] Zhao L, Ichise R, Sasaki Y, et al. Fast decision making using ontology-based knowledge base[C]//*Intelligent Vehicles Symposium (IV)*, 2016 IEEE. IEEE, 2016: 173–178.
- [9] Vacek S, Gindele T, Zollner J M, et al. Using case-based reasoning for autonomous vehicle guidance[C]//*Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*. IEEE, 2007: 4271–4276.
- [10] Chen C, Seff a, Kornhauser A, et al. Deepdriving: Learning affordance for direct perception in autonomous driving[C]//*Proceedings of the IEEE International Conference on Computer Vision*. 2015: 2722–2730.
- [11] Bouton M, Cosgun a, Kochenderfer M J. Belief state planning for autonomously navigating urban intersections[C]//*Intelligent Vehicles Symposium (IV)*, 2017 IEEE. IEEE, 2017: 825–830.
- [12] Russal S, Norvig P. Artificial intelligence: a modern approach[M]. Pearson Education, 2002.

- [13] Furda a, Vlacic L. Enabling Safe Autonomous Driving in Real-World City Traffic Using Multiple Criteria Decision Making[J]. IEEE Intelligent Transportation Systems Magazine, 2011, 3(1): 4–17.
- [14] Bai H, Cai S, Ye N, et al. Intention-aware online POMDP planning for autonomous driving in a crowd[C]//2015 IEEE International Conference on Robotics and Automation (ICRA). 2015: 454–460.
- [15] Wei J, Snider J M, Gu T, et al. A behavioral planning framework for autonomous driving[C]//2014 IEEE Intelligent Vehicles Symposium Proceedings. 2014: 458–464.
- [16] Aoude G S, Luders B D, How J P, et al. Sampling-Based Threat Assessment Algorithms for Intersection Collisions Involving Errant Drivers[J]. 2010.
- [17] Martin A, Schenato L, Althoff D. Interactive Motion Prediction using Game Theory[J]. : 100.
- [18] Vitelli M, Nayebi a. Carma: A Deep Reinforcement Learning Approach to Autonomous Driving[J]. 2016.
- [19] Gonzalez D, Perez J, Milanes V, et al. A Review of Motion Planning Techniques for Automated Vehicles[J]. IEEE Transactions on Intelligent Transportation Systems, 2016, 17(4): 1135–1145.
- [20] WAYDO S, MURRAY R M. Vehicle motion planning using stream functions[C]//2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422). 2003, 2: 2484–2491 vol 2.
- [21] DAILY R, BEVLY D M. Harmonic potential field path planning for high speed vehicles[C]//2008 American Control Conference. 2008: 4609–4614.
- [22] The DARPA Urban Challenge: autonomous vehicles in city traffic[M]. BUEHLER M, IAGNEMMA K, SINGH S. Berlin: Springer, 2009.

