智能汽车路径规划与轨迹跟踪系列算法精讲及Matlab程序实现

创作者：Ally

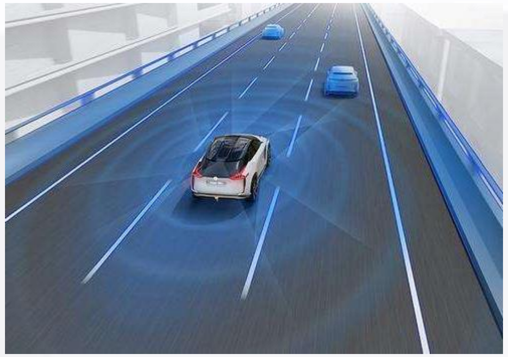
时间：2020/12/28

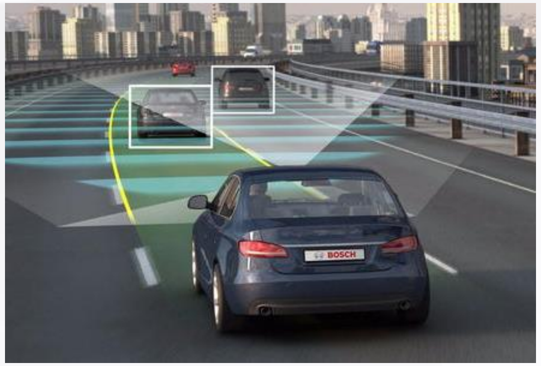
1 学习课程大纲目录

- **路径规划与轨迹跟踪系列学习视频**
  - 自动驾驶汽车定位-感知-规划-决策-控制概述
  - 全局路径规划
    - Dijkstra算法
    - 蚁群算法
    - 动态规划算法
    - A*算法
  - 局部路径规划
    - 多项式曲线法
    - 势场法
    - 贝塞尔曲线
    - B样条曲线
  - 轨迹跟踪与控制
    - 纯跟踪法
    - Stanley法
    - PID法
    - MPC法

1 自动驾驶汽车定位-感知-规划-决策-控制概述

定位 → 感知 → 规划 → 决策控制





定位

- 定位，即通过GPS、惯导、激光雷达等传感器，获取车辆的位置和航向信息。
- 绝对定位是指通过GPS实现，采用双天线，通过卫星获得车辆在地球上的绝对位置和航向信息。
- 相对定位是指根据车辆的初始位姿，通过惯导、里程计等传感器获得加速度和角加速度信息，将其对时间进行积分，即可得到相对初始位姿的当前位姿信息。

感知

- 环境感知，即通过摄像头、激光雷达、毫米波雷达、超声波雷达等多种传感器，感知周围的环境信息和车辆状态信息。
- 环境信息包括：道路、方向、曲率、坡度、车道，交通标志，信号灯；车辆状态信息包括：车辆的前进速度、加速度、转向角度、车身位置及姿态等。
- 多种传感器虽然可以获得丰富、细致的环境信息，但如何对多种传感器的信息进行融合统一处理，

规划

- 规划是对未来时域、空域的车辆一系列动作的计划。从涉及的时空大小分为全局（宏观）路径规划和局部（微观）路径规划。
- 全局路径规划指在已知全局地图的情况下，从车辆当前位置规划出一条到目的地的全局路径。
- 局部路径规划指根据环境感知的信息在换道、转弯、躲避障碍物等情况下，实时规划出一条安全、平顺、舒适的行驶路径。，

决策控制

- 决策控制，包括决策和控制两部分。
- 决策，在整个无人驾驶系统中，扮演者"驾驶员大脑"的角色，根据定位、感知及路径规划的信息，决定无人车的形式策略。包括：选取哪条车道、是否换道、是否跟车行驶、是否绕行、是否停车等。
- 控制，主要包括转向、驱动、制动三方面的控制，执行规划决策模块下发的期望速度和期望转向角度，也包括转向灯、喇叭、门窗等的控制。

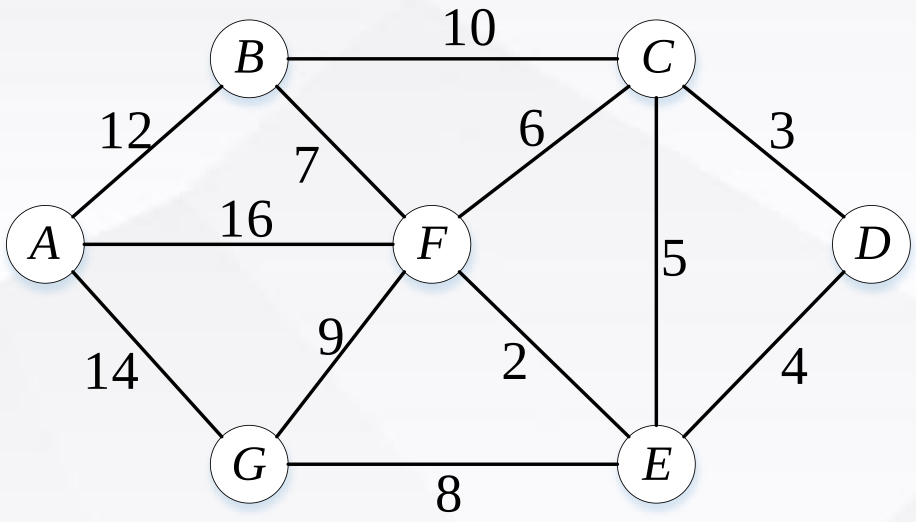
全局路径规划算法——Dijkstra算法

算法简介

- 迪杰斯特拉算法(Dijkstra)是由荷兰计算机科学家狄克斯特拉于1959 年提出的，因此又叫狄克斯特拉算法。是从一个节点遍历其余各节点的最短路径算法，解决的是有权图中最短路径问题。

算法思想

- 设G=(V,E)是一个带权有向图，把图中节点集合V分成两组，第一组为已求出最短路径的节点集合（用S表示，初始时S中只有一个源点，以后每求得一条最短路径，就将该节点加入到集合S中，直到全部节点都加入到S中，算法就结束了）；
- 第二组为其余未确定最短路径的节点集合（用U表示），按最短路径长度的递增次序依次把第二组的节点加入S中。在加入的过程中，总保持从源点v到S中各节点的最短路径长度不大于从源点v到U中任何节点的最短路径长度。
- 此外，每个节点对应一个距离，S中的节点的距离就是从v到此节点的最短路径长度，U中的节点的距离，是从v到此节点只包括S中的节点为中间节点的当前最短路径长度。



节点的邻近节点表

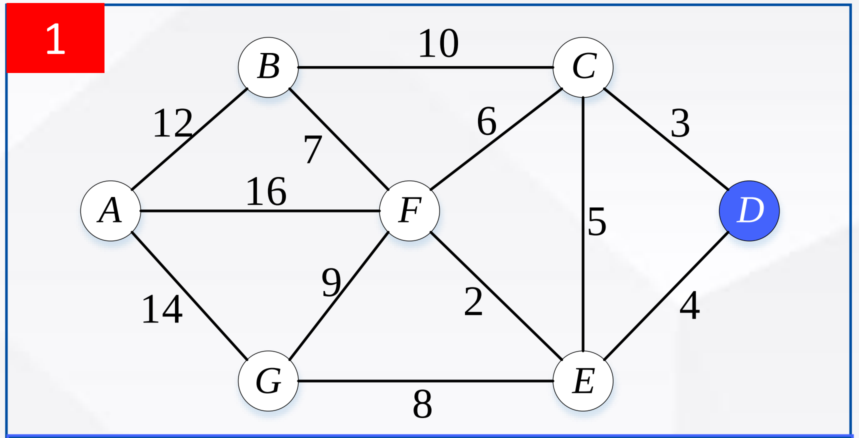
字母节点	A	B	C	D	E	F	G
邻节点	B/F/G	A/C/F	B/D/E/F	C/E	C/D/F/G	A/B/C/E/G	A/E/F

字母节点-数字节点对应表

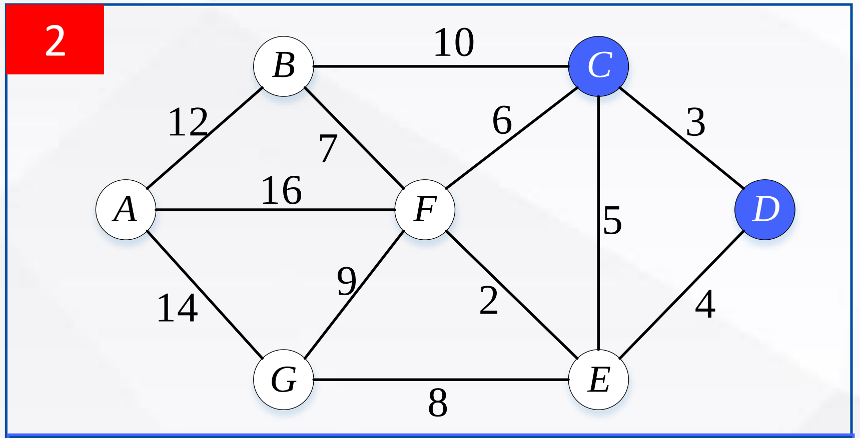
字母节点	A	B	C	D	E	F	G
数字节点	1	2	3	4	5	6	7

算法精讲

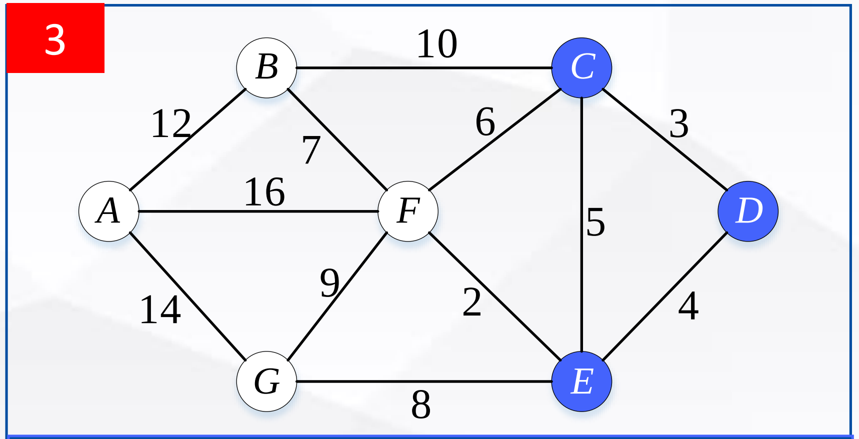
- 初始时，S只包含起点s；U包含除s外的其他节点，且U中节点的距离为"起点s到该节点的距离"[例如，U中节点v的距离为(s,v)的长度，然后s和v不相邻，则v的距离为∞]。
- 从U中选出"距离最短的节点k"，并将节点k加入到S中；同时，从U中移除节点k。
- 更新U中各个节点到起点s的距离。之所以更新U中节点的距离，是由于上一步中确定了k是求出最短路径的节点，从而可以利用k来更新其它节点的距离；例如，(s,v)的距离可能大于(s,k)+(k,v)的距离。
- 重复步骤(2)和(3)，直到遍历完所有节点。



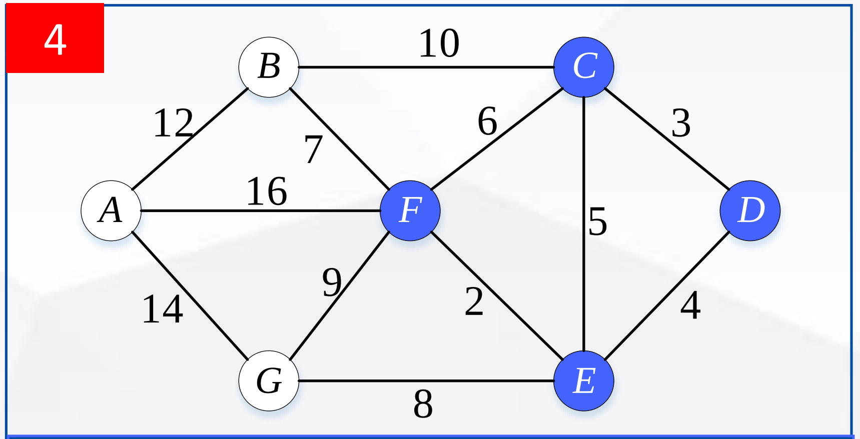
选取节点D，S={D(0)}
U={A(∞), B(∞), C(3), E(4), F(∞), G(∞)}



选取节点C， S={D(0), C(3)}
U={A(∞), B(13), E(4), F(9), G(∞)}



选取节点E，S={D(0), C(3), E(4)}
U={A(∞), B(13), F(6), G(12)}



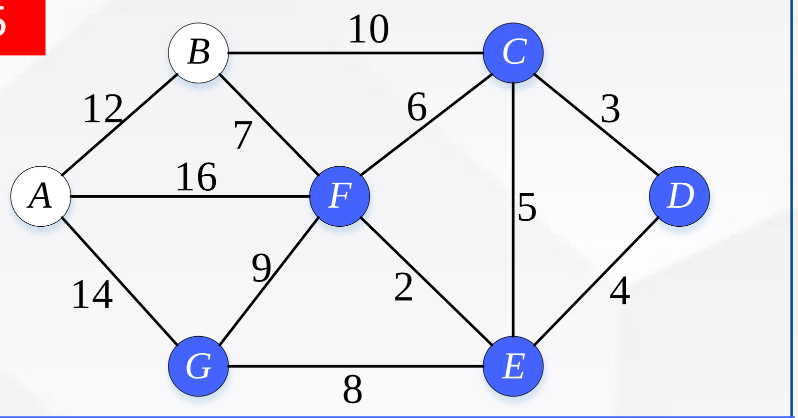
选取节点F，S={D(0), C(3), E(4), F(6)}
U={A(22), B(13), G(12)}

2 全局路径规划算法——Dijkstra算法

算法精讲

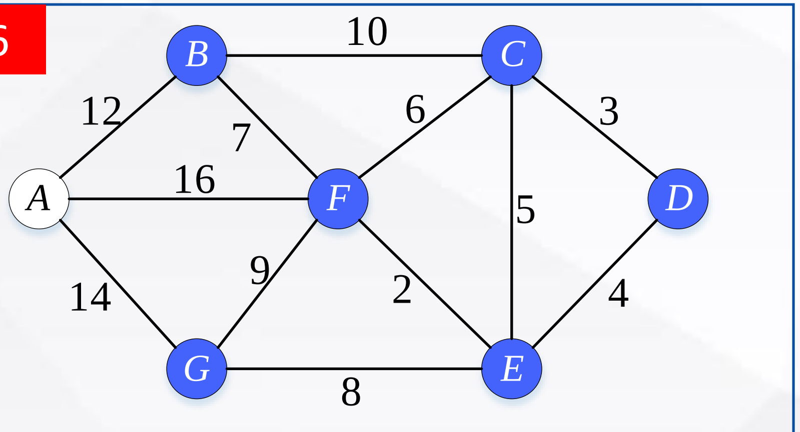
- 初始时，S只包含起点s；U包含除s外的其他节点，且U中节点的距离为"起点s到该节点的距离"[例如，U中节点v的距离为(s,v)的长度，然后s和v不相邻，则v的距离为∞]。
- 从U中选出"距离最短的节点k"，并将节点k加入到S中；同时，从U中移除节点k。
- 更新U中各个节点到起点s的距离。之所以更新U中节点的距离，是由于上一步中确定了k是求出最短路径的节点，从而可以利用k来更新其它节点的距离；例如，(s,v)的距离可能大于(s,k)+(k,v)的距离。
- 重复步骤(2)和(3)，直到遍历完所有节点。

5



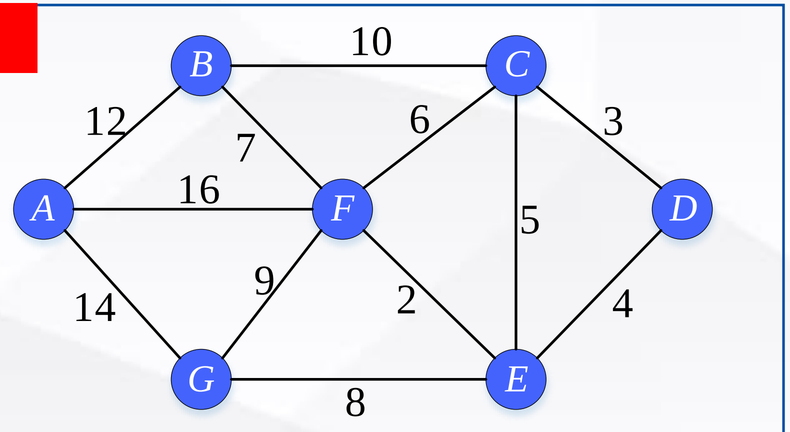
S={D(0), C(3), E(4), F(6), G(12)}
U={A(22), B(13)}

6



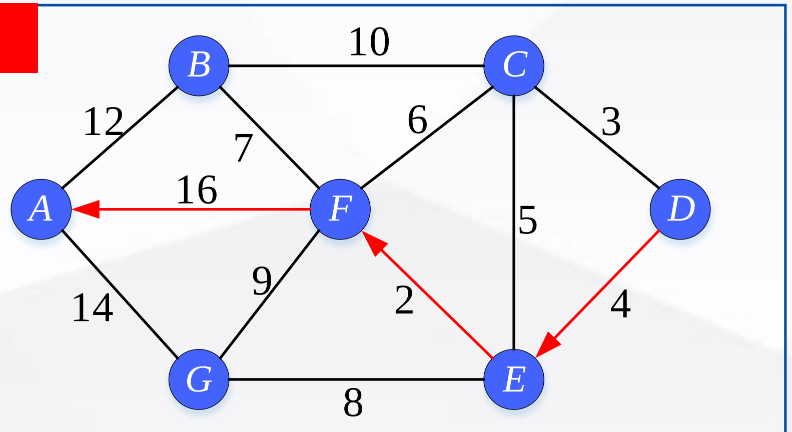
S={D(0), C(3), E(4), F(6), G(12), B(13)}
U={A(22)}

7



S={D(0), C(3), E(4), F(6), G(12), B(13), A(22)}
U=∅

8



D→A的最优路径为D→E→F→A
最短距离为22

谢谢大家