



自动驾驶汽车 预测-决策-规划-控制实战入门

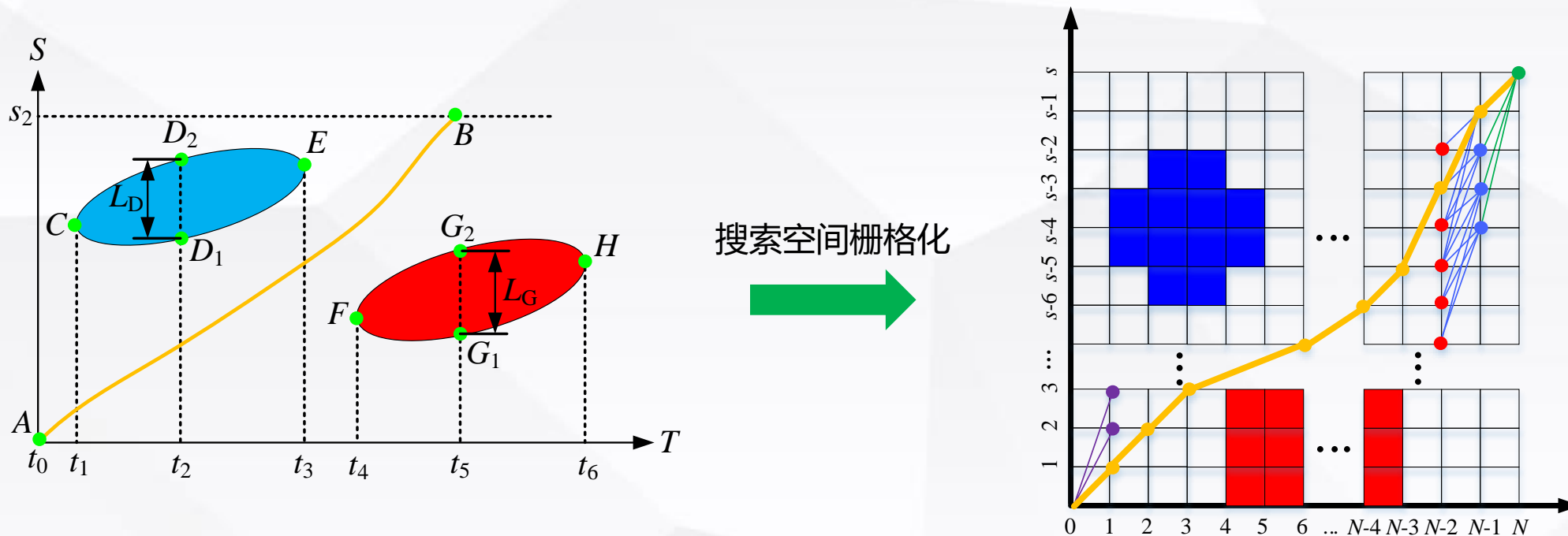
5.2 基于动态规划算法初步规划ST曲线

创作者: Ally

时间: 2021/11/21



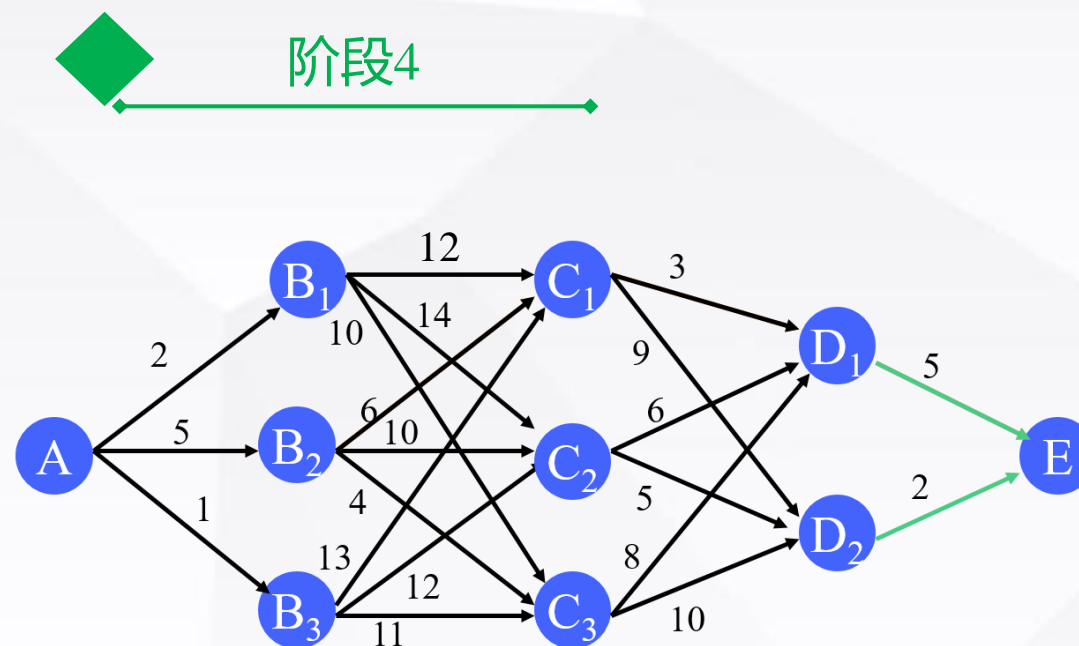




ST图可行搜索空间的栅格化处理示意图

- ◆ 蓝色和黄色区域体现了周边交通车在本车规划轨迹上的占据情况，可将其视为禁忌搜索区域。
- ◆ 当确定某个目标函数后，最优速度规划就是以ST图的禁忌搜索区域作为一种约束条件，在可行域内搜索一条可行且最优的速度序列。这类问题本质上是一个多阶段最优决策问题，可以采用**动态规划 (Dynamic Programming, DP)** 的思想进行求解。

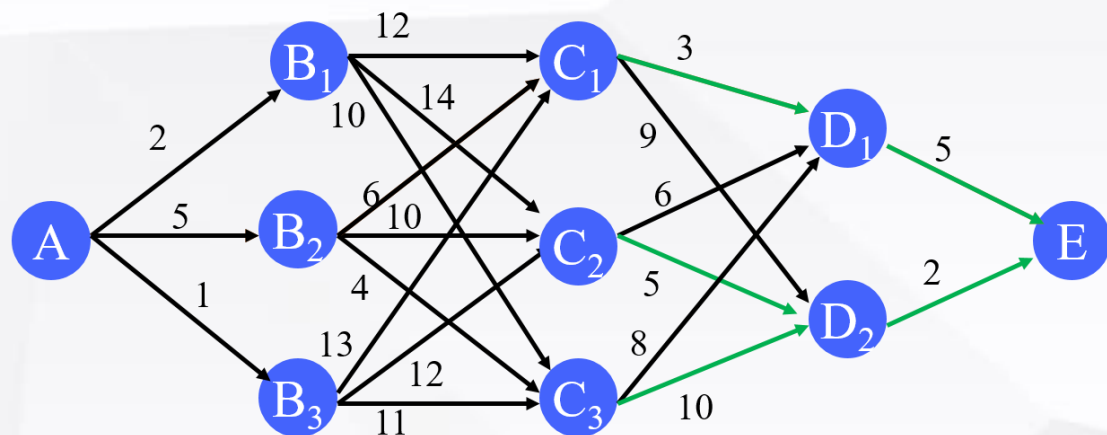
- ◆ 通常用逆向寻优，正向求解的思路
- ◆ DP算法本质由三层循环构成：
 - 第一层：由右到左依次遍历每一个阶段；
 - 第二层：遍历第*i*个阶段的每一个状态；
 - 第三层：遍历第*i*+1个阶段的每一个状态。



➤ 第四阶段 ($D \rightarrow E$) : D 有两条路线到终点 E 。

$$f_4(D_1) = 5 \quad f_4(D_2) = 2$$

阶段3



- 第三阶段 (C → D) : C 到D 有 6 条路线。
- 第3阶段的C有3个状态值, 分别讨论经过该状态值的最优路线。

经过C1

$$f_3(C_1) = \min \begin{cases} d(C_1, D_1) + f_4(D_1) \\ d(C_1, D_2) + f_4(D_2) \end{cases} = \min \begin{cases} 3 + 5 \\ 9 + 2 \end{cases} = 8$$

➤ 最短路线为C1→D1 →E

经过C2

$$f_3(C_2) = \min \begin{cases} d(C_2, D_1) + f_4(D_1) \\ d(C_2, D_2) + f_4(D_2) \end{cases} = \min \begin{cases} 6 + 5 \\ 5 + 2 \end{cases} = 7$$

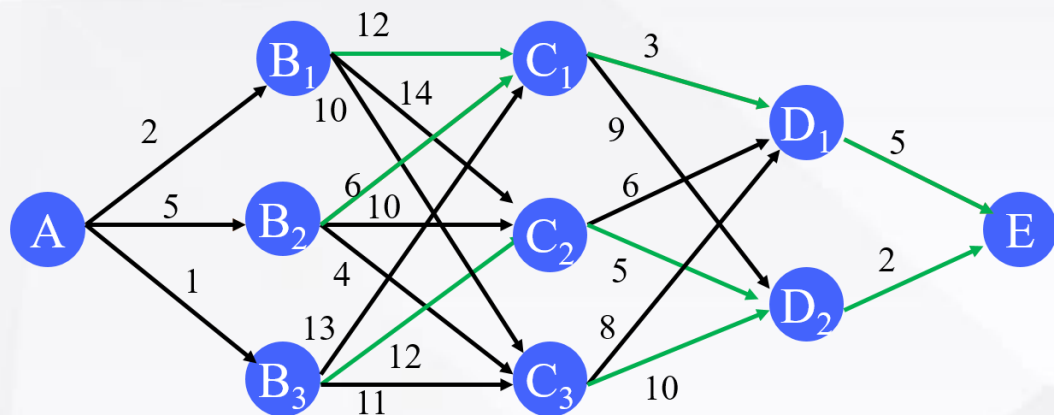
➤ 最短路线为C2→D2 →E

经过C3

$$f_3(C_3) = \min \begin{cases} d(C_3, D_1) + f_4(D_1) \\ d(C_3, D_2) + f_4(D_2) \end{cases} = \min \begin{cases} 8 + 5 \\ 10 + 2 \end{cases} = 12$$

➤ 最短路线为C3→D2 →E

阶段2



- ◆ 第二阶段 (B → C) : B 到C 有 9 条路线。
- ◆ 第2阶段的B有3个状态值, 分别讨论经过该状态值的最优路线。

经过B1

$$f_2(B_1) = \min \begin{cases} d(B_1, C_1) + f_3(C_1) \\ d(B_1, C_2) + f_3(C_2) \\ d(B_1, C_3) + f_3(C_3) \end{cases} = \min \begin{cases} 12 + 8 \\ 14 + 7 \\ 10 + 12 \end{cases} = 20$$

➤ 最短路线为B1→C1→D1→E

经过B2

$$f_2(B_2) = \min \begin{cases} d(B_2, C_1) + f_4(C_1) \\ d(B_2, C_2) + f_4(C_2) \\ d(B_2, C_3) + f_4(C_3) \end{cases} = \min \begin{cases} 6 + 8 \\ 10 + 7 \\ 4 + 12 \end{cases} = 14$$

➤ 最短路线为B2→C1→D1→E

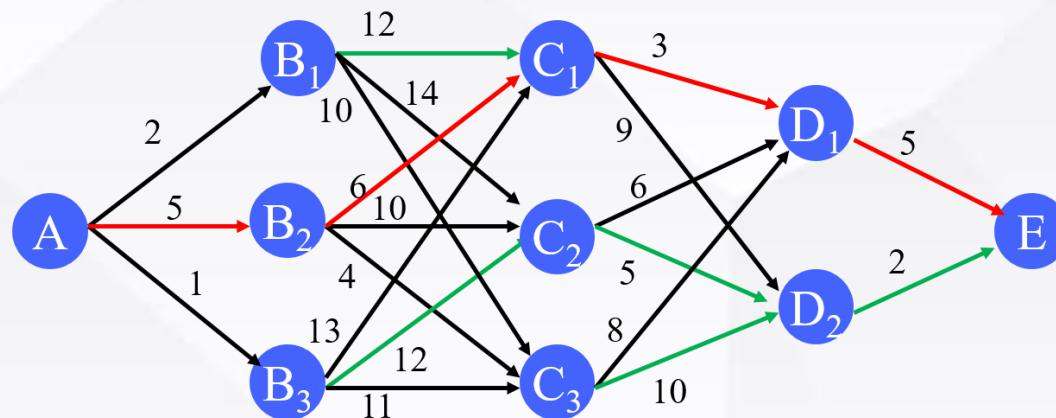
经过B3

$$f_2(B_3) = \min \begin{cases} d(B_3, C_1) + f_3(C_1) \\ d(B_3, C_2) + f_3(C_2) \\ d(B_3, C_3) + f_3(C_3) \end{cases} = \min \begin{cases} 13 + 8 \\ 12 + 7 \\ 11 + 12 \end{cases} = 19$$

➤ 最短路线为B3→C2→D2→E



阶段1



➤ 第一阶段 (A → B) : A 到 B 有 3 条路线。

$$f_1(A) = \min \begin{cases} d(A, B_1) + f_2(B_1) \\ d(A, B_2) + f_2(B_2) \\ d(A, B_3) + f_2(B_3) \end{cases} = \min \begin{cases} 2 + 20 \\ 5 + 14 \\ 1 + 19 \end{cases} = 19$$

➤ 最短路线为 A → B2 → C1 → D1 → E

- ◆ 关于“逆向寻优，正向求解”的进一步思考：
 - 当最后一个阶段的状态确定且唯一时，一般用此顺序；
 - ST图并未要求最后一个阶段的s值，故直接用正向寻优即可。
- ◆ 由于动态规划的本质是三层循环的遍历，需要合理设定栅格化颗粒度：
 - 较粗糙的颗粒度会使得曲线偏离最优；
 - 较精细的颗粒度会使得运算时间延长。
- ◆ 编程时，需要灵活运用MATLAB的cell类型、mat类型的变量，以方便存储、调用动态规划过程中产生的大量数据。