

自动驾驶决策规划算法第二章第四节(上)

二次规划

求解使二次型 $\frac{1}{2} x^T H x + f^T x$ 值最小的 x

并且满足约束 ~~$lb \leq Ax \leq ub$~~ $Ax \leq ub$

($Aeq x = beq$, $Ax \leq b$, $lb \leq x \leq ub$) $\Rightarrow A_1 x \leq b_1$

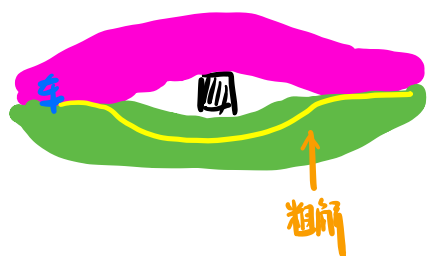
$$lb \leq x \leq ub \Rightarrow \begin{cases} x \leq ub \\ x \geq lb \end{cases} \Rightarrow \begin{matrix} x \leq ub \\ -x \leq -lb \end{matrix} \Rightarrow \begin{pmatrix} I \\ -I \end{pmatrix} x \leq \begin{pmatrix} ub \\ -lb \end{pmatrix}$$

$$Aeq x = beq \Rightarrow beq \leq Aeq x \leq beq \Rightarrow \begin{cases} Aeq x \leq beq \\ -Aeq x \leq -beq \end{cases} = \begin{pmatrix} Aeq \\ -Aeq \end{pmatrix} x \leq \begin{pmatrix} beq \\ -beq \end{pmatrix}$$

$$\begin{pmatrix} A \\ I \\ -I \\ Aeq \\ -Aeq \end{pmatrix} x \leq \begin{pmatrix} b \\ ub \\ -lb \\ beq \\ -beq \end{pmatrix} \Rightarrow A_1 x \leq b_1$$

无需在意二次规划的约束形式 ($lb \leq Ax \leq ub$, $a \leq x \leq b$, $Ax \leq b$)

二次规划要求解空间是凸的, 而动态规划开辟了凸空间



粗解落在了哪个凸空间上, 那么最终解就用它作为解空间

动态规划与决策的关系

决策算法 $\begin{cases} \text{重决策} \longrightarrow \text{基于人给定的规则} \\ \text{轻决策} \longrightarrow \text{基于代价函数} \end{cases}$

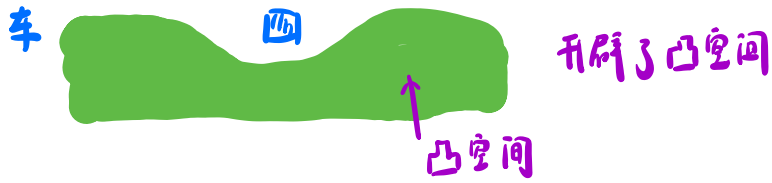
车

四

重决策: 根据人给定的规则判断 (综合与障碍物的距离, 相对速度, 周围环境等信息) 给出决策

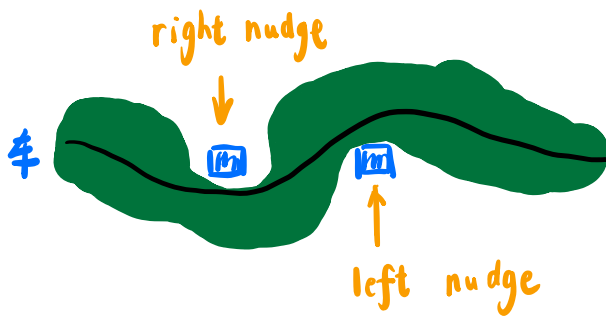
人事先写好 场景与决策的映射关系

假设决策: right nudge (向右绕)



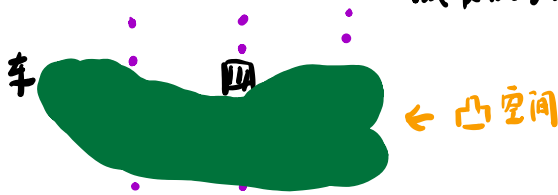
缺点: ① 场景太多了, 人无法完全覆盖

② 人给出的决策所开辟的凸空间未必满足约束



凸空间约束过于严格, 二次规划搜不到满足动力学约束的解

轻决策: 无先验规则, 空间离散化, 设计 cost function, 动态规划算法求前离散空间的最优路径
该最优路径开辟凸空间



优点: ① 无人为规则, 可以处理复杂场景

② 有粗解, 通过设计代价函数, 可以使粗解“满足”硬约束 (碰撞, 最小曲率), 这样使二次规划求解成功的几率大大增加 (因为粗解在凸空间内部, 所以该凸空间的“扭曲”程度至少有粗解为底) 大大缓解了基于人为规则的决策所造成的凸空间扭曲情况

轻决策缺点: 复杂场景计算量很大

依赖预测 (路径规划时会用到)

要求周围环境全知 (感知, 定位要求高)

代价函数设计 / 最优解未必符合人的驾驶习惯

重决策优点: 计算量小

在感知不强的情况下仍然能做决策 (融合人的智慧)

L2+ 高速 L3 \Rightarrow 重决策

2022.3

L4 \Rightarrow 轻决策

博弈问题 (目前的热点) Apollo 7.0 预测