实验二

车联网环境下一个基于博弈论的十字路口智能交通控制算法

1. 车联网环境

交通拥堵日益成为众多大城市所面临的问题，十字路口是导致交通拥堵的主要瓶颈之一（大部分的交通拥堵来自十字路口红灯的等待）。但在自动驾驶全面普及的未来，这一问题有望得到解决。现有状态控制系统（例如停车标志，交通信号等）将被取代，自动车辆均配备协同自适应巡航控制（CACC）系统，交通管理系统通过控制每一辆车的行进状态来使交通通行量得到最优化。本次实验内容——车联网环境下一个基于博弈论的十字路口交通控制算法，即该应用的核心技术之一。

在车联网环境下，车辆可以与交叉路口的中央管理中心通信，以提供其瞬时速度，位置和方向。在中央管理中心融合所有的信息之后，决定每辆车的行动（加速，减速或保持当前速度），以避免碰撞并为每辆车提供最低延迟。

2. 胆小鬼博弈

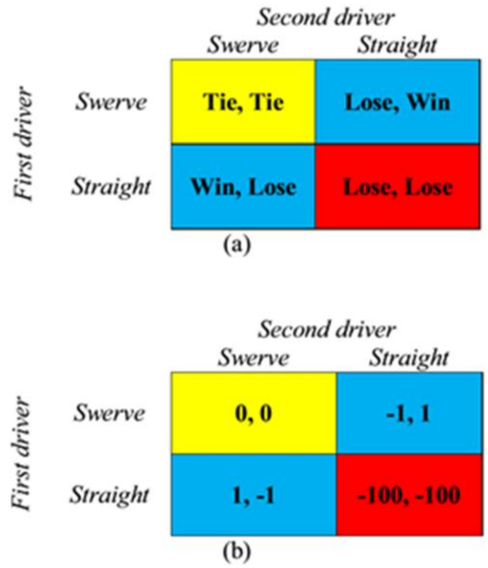

图 1. 胆小鬼博弈

胆小鬼博弈（英文：The game of chicken），又译懦夫博弈，是博弈论中一个影响深远的模型，逻辑就是“不要命的最大”。模型中，两名车手相对驱车而行，谁最先转弯的一方被耻笑为“胆小鬼”（chicken），让另一方胜出，因此这博弈模型在英文中称为 The Game of Chicken（懦夫游戏），但如果两人拒绝转弯，任由两车相撞，

最终谁都无法受益。其收益矩阵如图.1 所示。本方法受胆小鬼博弈启发。

- 玩家

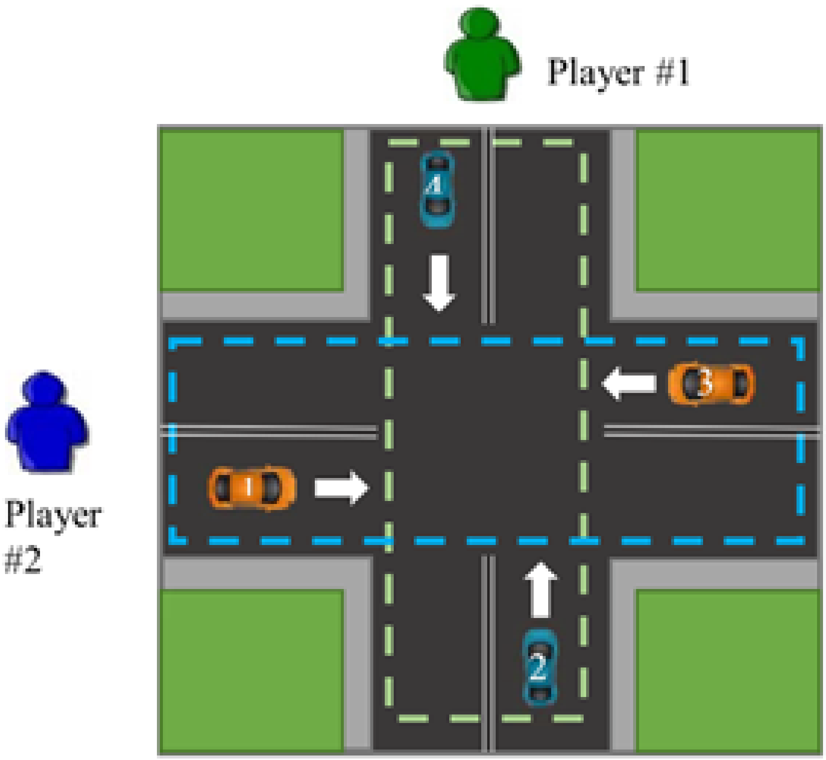

图 2. 博弈交通智能控制

该博弈只有两名玩家。 $Player\#1$决定车辆 2 和 4 的动作，而$Player\#2$决定车辆 1 和 3 的动作，如图 2 所示。每个玩家想要以这样的方式控制车辆以最小化它们在交叉路口的延迟并保证它们安全地（未发生碰撞）穿过十字路口。

- 玩家行动

每个玩家最三种可能的动作：加速（表示为1），减速（表示为-1）或以当前速度继续行驶（表示为0）。由于每个玩家有两辆车并且每辆车都有动作组，因此玩家的动作是车辆动作的笛卡尔乘积。每辆汽车可以采取的动作是$\{1,\ 0, -1\}$，则每个玩家可以采取的动作有 9 种，即$\{1,0,-1\}\times\{1,0,-1\} = \{(1,1),(1,0),(0,1),(1,-1),(-1,1),(0,0),(0,-1),(-1,0),(-1,-1)\}$，例如$player\#1$的动作为$(1,0)$代表他命令 2 号车加速，3 号车保持当前速度行驶。

3. 在给定车辆数据与玩家策略后可以运行驾驶模拟

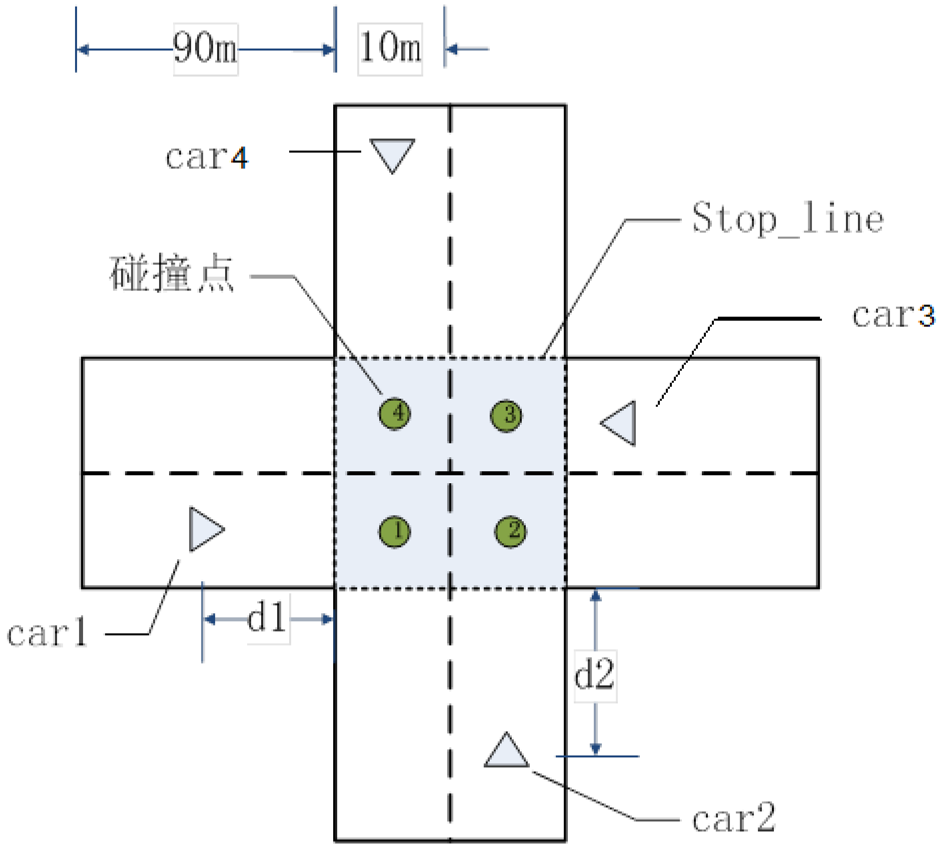

图 3：道路结构

3.1 数据：

道路结构如图 3 所示。每次模拟中随机生成 4 辆车的数据，每辆车的数据的形式为(d,v)，如图所示，d为车辆距离停车线 stop _line 的距离，范围为[0,90]，v为车辆的当前速度,初始化范围为[20,40]，规则是加速后速度不能超过 40，减速下限无限制。

3.2 根据车辆动作速度更新：

为简化计算过程，实验中加速（$\mathrm{action}=1$）直接将原速度提高 40%，减速（$\mathrm{action}=-1$）直接将原速度减少 40%，保持当前速度则速度不变。速度的更新公式为：

$$v\_new = v \times (1 + action \times 0.4) \tag{1}$$

3.3碰撞判定：

以图 3 中 4 号碰撞点为例，计算 3 号车与 4 号车到达碰撞点 4 的时间差，如时间差小于 1s 则判定为碰撞，否则判定为通过。

3.4博弈损失（通行时间）

一次模拟中 4 个碰撞点中有一个发生了碰撞则为博弈中双输的情况，该次博弈损失为无穷大（inf）。如未发生碰撞，则博弈损失为 4 辆车通过该十字路口的平均时间。一辆车(d, v)的通行时间为：

$$t = \begin{cases} \dfrac{90-d}{v} + \dfrac{110+d}{v_{new}} & if\ 未发生碰撞 \\ \mathrm{inf} & if\ 发生碰撞 \end{cases}$$

此处假设车辆经过一次速度调整后，在驶出路口前不再调整速度。

3.5计算实例

四辆车的数据分别为

$$\mathrm{car}_1 = (45, 23)$$

$$\mathrm{car}_2 = (71, 24)$$

$$\mathrm{car}_3 = (47, 28)$$

$$\mathrm{car}_4 = (76, 28)$$

- 玩家 1 的行动为（1，1），玩家 2 的行动为（1，0），注意玩家 1 控制的是车辆 2 与 4，注意玩家 2 控制的是车辆 1 与 3。

  1) 速度更新：

$$v1_{new} = 23 \times (1 + 1 \times 0.4) = 32.20$$

$$v2_{new} = 24 \times (1 + 1 \times 0.4) = 33.60$$

$$v3_{new} = 28 \times (1 + 0 \times 0.4) = 28.00$$

$$v4_{new} = 28 \times (1 + 1 \times 0.4) = 39.20$$

2) 碰撞判断

以 4 号碰撞点（3 号车与 4 号车）为例：

$$\mathrm{t3} = (47 + 15)/\ 28.00 = 2.21$$

$$\mathrm{t4} = (76 + 5)/\ 39.20 = 2.06$$

时间差小于 1s，发生碰撞，此次模拟通行时间为无穷大 inf。

- 若玩家 1 的行动为（0，0），玩家 2 的行动为（1，1），进行新的模拟：

1) 速度更新：

$$v1_{new} = 23 \times (1 + 1 \times 0.4) = 32.20$$

$$v2_{new} = 24 \times (1 + 0 \times 0.4) = 24.00$$

$$v3_{new} = 28 \times (1 + 1 \times 0.4) = 39.20$$

$$v4_{new} = 28 \times (1 + 0 \times 0.4) = 28.00$$

2) 碰撞判断

以 1 号碰撞点（3 号车与 4 号车）为例：

$$\mathrm{t3} = (47 + 15)/\ 39.20 = 1.58$$

$$\mathrm{t4} = (76 + 5)/\ 28.00 = 2.89$$

时间差大于 1s，未发生碰撞。同理计算 1，2，3 号碰撞点可发现均未发生碰撞。

3) 计算通行时间：

1 号车的通行时间：

$$t_1 = \frac{90-45}{23} + \frac{110+45}{32.2} = 6.77$$

同理可计算出每一辆车的通行时间，进而可以求出 4 辆车的平均通行时间，即本次模拟的博弈损失。

4. 实验

4.1 计算博弈损失矩阵

四辆车的数据分别为

$$\mathrm{car}_1 = (45, 23)$$

$$\mathrm{car}_2 = (71, 24)$$

$$\mathrm{car}_3 = (47, 28)$$

$$\mathrm{car}_4 = (76, 28)$$

遍历所有的决策组合，计算出博弈损失矩阵（通行时间），并给出最快通过时的玩家的决策：

Player2 / Player1	(1,1)	…	(-1,-1)
(1,1)			
…			
(-1,-1)			

通行时间最短：

player#1 采取的策略 ：______

player#2 采取的策略 ：______

4.2 与传统方法对比

传统方法：先截断一个方向的车流，让另一个方向的车全部通行后在开流。例如先让车 2，4 通行，后让 1，3 通行。通行时间为

$$t = \max\left(\frac{200}{v\_1}, \frac{200}{v\_3}\right) + \max\left(\frac{200}{v\_2}, \frac{200}{v\_4}\right)$$

每次实验中，随机生成 4 辆车的数据，计算该数据在传统方法下的通行时间，再计算 4.1 中的矩阵，以矩阵中的最小值为博弈方法的通行时间。

进行 10 次实验，分别记录 10 次实验中博弈方法和传统方法的结果，书写实验报告。