

数学建模第二次作业论文

——基于元胞自动机的十字路口车流问题研究

15051036 刘星超

15231030 李子睿

摘要

十字路口绿灯时通过路口的车流量问题是与我们的生活息息相关的问题。本文基于元胞自动机方法，对在有限速情况下绿灯时车辆的行为模式进行了建模。接着，对十字路口由红灯转为绿灯时 30 秒内车流的情况，本文通过 MATLAB 编程进行了模拟，还统计了 30 秒内通过十字路口的车辆数。最后，本文对模拟结果进行了分析，总结出了限速与通过车辆数之间的关系，并提出一种提高车辆通过效率的措施。

关键词：元胞自动机 交通流 MATLAB

1 问题重述与分析

1.1 问题重述

本问题要求模拟在十字路口绿灯亮 30 秒的情况下汽车通过十字路口的过程，换言之，对汽车起步加速的过程和车辆之间的互动进行建模从而求出 30 秒内通过十字路口的车辆数。同时题目要求研究不同限速对十字路口车流量的影响，也就是不同的车辆最大速度对在 30 秒内通过十字路口的车辆数的影响。

1.2 难点与关键

由于题目要求使用元胞自动机，问题的关键主要在于对每个元胞进行建模。考虑到问题的实际背景，关键在于描述每辆车的加速度、速度关于时间的变化关系和每辆车之间的互动关系。

2 模型假设

2.1 车辆不变道，各车道互不干扰，车辆均在车道内行驶

“加塞”现象会严重的影响十字路口车辆的通行方式与通行量，而在不发生随意“加塞”的情况下，根据常识，机动车会沿着其所在车道直接加速向前。为了更直观的对绿灯状况下的十字路口通行进行模拟，本文将不考虑多车道之间的变道。

2.2 各车辆具有相同的尺寸和机动性能

在城市道路中，主要的行驶车辆就是车长在 3 米到 6 米的轿车。为了简化问题，本文假设车辆的车长均为 4 米。另外，对于城市行驶的状况，车辆之间性能的差距体现的并不明显。即便有些车的加速性能好，有些车的加速性能差，但是这些差距都可以通过驾驶员踩油门的力度来进行弥补。所以不妨认为它们的机动性能相同。

2.3 车辆启动前为紧密排列，车距为 0.5 米

根据常识，十字路口处红灯时车辆排列非常紧密，车辆之间距离很小，基于常识，车距设置为 0.5 米是一个合理的选择。

2.4 驾驶员驾驶方式一致且遵纪守法

驾驶员的驾驶风格迥异，具有随机性且难以模拟。在不影响问题的本质的情况下，本文认为每辆车的驾驶员驾驶方式相同，也就是每辆车的加速模式一致。另一方面，危险驾驶者是人群中的极少数，本文认为每位驾驶者都是遵纪守法的公民，也就是每辆车都不会超过限速。

2.5 不考虑碰撞等意外因素

2.6 元胞自动机每次状态更新的时间间隔是 1 秒

一方面，1 秒是在使用元胞自动机方法研究交通流的大量工作中被广泛使用的一个时间间隔；另一方面，文献中给出的驾驶员在驾驶时的反应时间也在 1 秒左右。综上，1 秒是一个合理的选择。

3 建立模型

3.1 符号表

| 含义 | 符号 |
|------------------|---------|
| 30 秒内通过的车辆总数 | n |
| 每辆车的车长 | l |
| 第 i 辆车的车头的初始坐标 | x_i^0 |
| 第 i 辆车的车头的当前坐标 | x_i |
| 第 i 辆车的当前速度 | v_i |
| 车辆的加速度 | a_i |
| 路段的最高限速 | v_m |

3.2 道路模型

考虑到城市交通的实际状况，本文将考察长度为 1 公里的单车道路段。在车辆均在车道内行驶的前提下，该路段可被抽象成 $[0, 1000]$ 这一段数轴。由于车辆通过十字路口后对十字路口的车流量影响极小，故本文将十字路口的起点（也就是停车线的位置）设置在 700 米处（而不是 500 米处）。也就是说，一旦元胞的车头坐标 x_i 从小于等于 700 变化到大于 700，本文就认为通过十字路口的车辆数增加 1。

3.3 车辆模型

3.3.1 车辆的属性

本文按如下顺序给车辆编号：当红灯时，车辆紧密排列，离停车线最近的车编号是 1，第二近的车编号是 2，以此类推。考虑到时间上限为 30 秒，本文将对 100 辆车进行模拟。

第 i 辆车用如下四个参数描述：（1）车头的初始坐标 x_i^0 （2）车头的当前坐标 x_i （3）车辆的当前速度 v_i （4）车辆的加速度 a_i

3.3.2 车辆车头的初始坐标

按照常识，等红灯时离停车线最近的车，也就是 1 号车，会将车头尽可能靠近停车线，两者之间距离可以忽略，由此得到： $x_1^0 = 700$ 。考虑到每辆车的车长是 4 米，车间距是 0.5 米，可以得到 $x_i^0 = 700 - 4.5(i - 1)$ 。

3.3.3 车辆的加速度、速度和车头坐标

查阅文献知，十字路口车辆起步可以近似为匀加速运动，也就是加速度不变的直线运动，加速度的平均值在 2m/s^2 到 2.5m/s^2 之间，本文取全部车辆的加速度 a_i 为 2m/s^2 。这样得到，在第 i 辆车启动行驶 t 秒后，其速度是 $2t$ m/s（当该速度不超过限速时）。若 $2t > v_m$ ，则车辆速度保持 v_m 不变。

根据物理知识可以知道，在第 i 辆车启动行驶 t 秒后，车头坐标可表示为 $x_i(t) = x_i(t - 1) + (v_i(t - 1) + v_i(t))/2$ 。

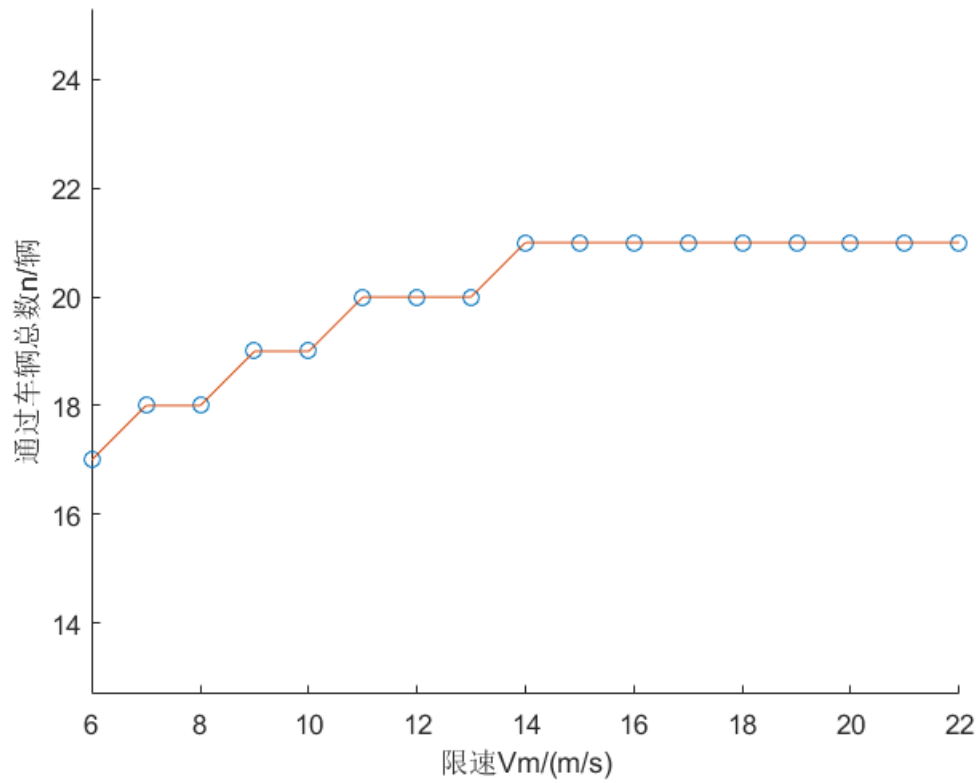
在车辆由匀加速运动转为以极限速度匀速运动的那一秒，车头坐标为 $x_i(t) = x_i(t - 1) + (v_i(t - 1) + v_m)(v_m - v_i(t - 1))/2a_i + v_m(1 - \frac{1}{a_i}(v_m - v_i(t - 1)))$ 。

3.3.4 车辆的行为模式

绿灯亮起后，第一辆车立刻起步进行匀加速运动。考虑到司机的反应时间，第 i 辆车将在第 $i-1$ 辆车起步 1 秒后启动，匀加速运动直到速度达到限速。

4 计算结果

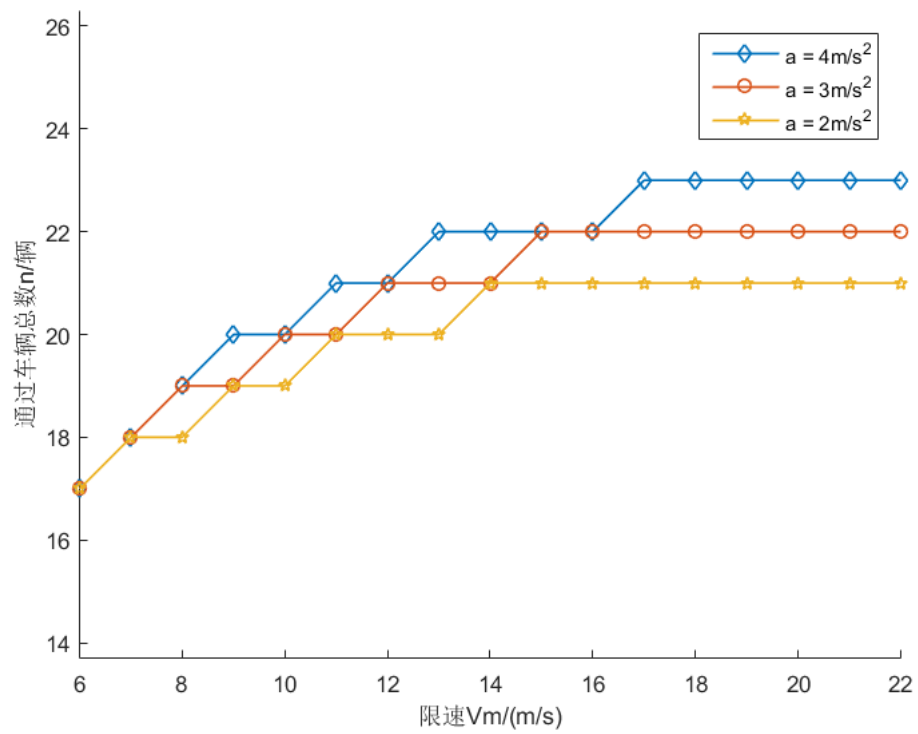
考虑到城市车辆限速一般在20km/h到80km/h之间，本文对最高限速从6m/s到22m/s用 MATLAB 编程模拟，结果如下图所示



5 验证可靠性及灵敏度分析

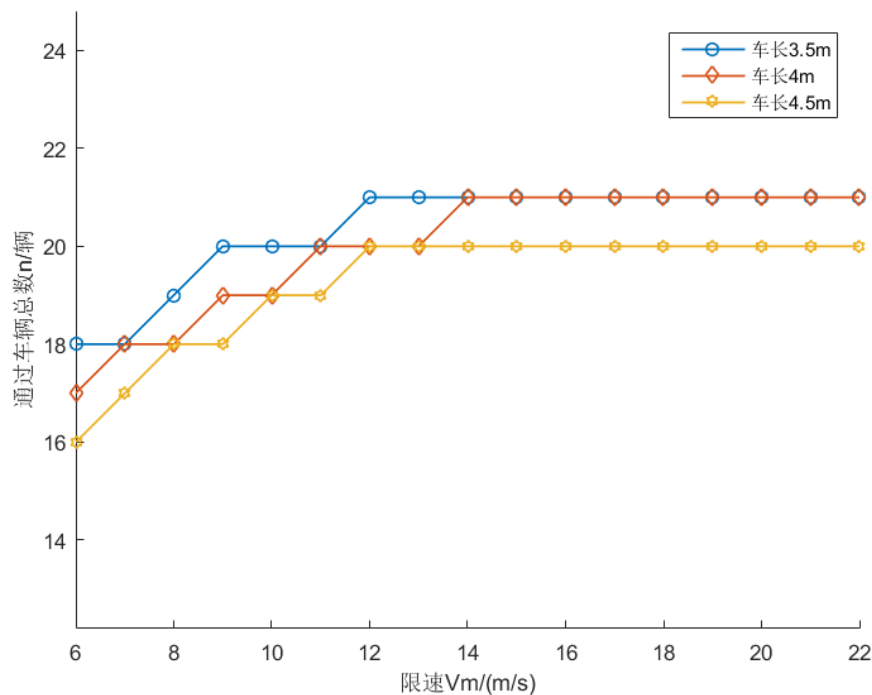
5.1 汽车加速度对模型的影响

分别令汽车加速度为 3m/s^2 和 4m/s^2 再次进行模拟，结果如下图所示。



随着加速度的增加，30 秒内通过十字路口的车辆总数有明显增加，符合常识，证明了本文中的模型是合理且有效的。

5.2 车辆长度对模型的影响



如折线图所示，当车长降低时，30 秒内通过的车辆数增高；当车长升高时，30 秒内通过的车辆数减少。这个结果与事实情况相符，体现了本文模型的正确性。

5.3 对计算结果可靠性和准确性的说明

日常生活的经验表明，十字路口绿灯时通过的车辆数主要取决于司机的反应时间和车辆的加速能力，而与车长的关系不大。从上面三幅折线图可以看出，加速度的增加显著的增加了通过的车辆数，而车长的缩短对通过车辆数的提高并不明显，恰恰与日常经验符合，说明了模型的可靠性。

另一方面，受限于司机的反应能力，30 秒内可以通过路口的车辆总数是有限的(实际上，就本题的假设而言，一个显然的车辆通过数上限是 30 辆)；同时，可以想见，受限于加速距离与加速能力，限速的提高对通过车辆数的提高是有瓶颈的。三幅折线图中，随着限速的不断提高，通过的车辆总数都达到一个定值不变，与分析相符，体现了模型的正确性与鲁棒性。

6 结论

本文基于元胞自动机模型，通过 MATLAB 编程模拟对通过十字路口的交通流进行了研究，得出了如下结论：

(1) 提高限速可以增加 30 秒内通过十字路口的车辆总数，但当限速大到一定地步后，对通过十字路口的车辆总数就没有影响了。

(2) 如果不考虑对安全性的影响，可以采用提高车辆加速度的方式来显著增加十字路口的车辆通过效率。

参考文献

- [1]徐学珍, 田宏伟. 快速路驾驶员反应时间的确定[J]. 信息技术与信息化, 2007(4):105-108.
- [2]贾常明, 常安德. 汽车加速度测量及在交通事故鉴定中的应用研究[J]. 中国刑警学院学报, 2013(2):41-42.

附录

此处的 MATLAB 代码是在加速度为 2m/s^2 ，车长为 3.5m ，限速为 11m/s 的情况时的 MATLAB 代码。

```
clear;

STOP = 0;

READY = 1;

GO = 2;

PASSED = 1;

NOTPASSED = 0;

vMax = 11;

acc = 2;

flag = 0;

time = 0;

for i = 1:100
    car(i, 1) = 700 - (i-1)*4;
    car(i, 2) = 0;
    car(i, 3) = STOP;
    car(i, 4) = NOTPASSED;
end

carAmount = 0;

car(1, 3) = GO;
```

```

for i = 1:30
    for j = 1:100
        flag = 0;

        if (car(j,3) == READY)
            car(j,3) = GO;
        end

        if ((j>1)&&(car(j,3) == STOP)&&(car(j-1,3) == GO))
            car(j,3) = READY;
        end

        if (car(j,3) == GO)
            vOld = car(j,2);

            if (vOld == vMax)
                flag = 2;
            end

            car(j,2) = acc + car(j,2);

            if (car(j,2) >= vMax)
                car(j,2) = vMax;

                if (flag == 0)
                    flag = 1;
                end
            end
        end

        vAvg = (vOld + car(j,2))/2;

        if (flag == 1)
            time = (vMax - vOld)/acc;
        end
    end
end

```

```
car(j,1) = car(j,1) + time*(vOld+vMax)/2 +  
vMax*(1-time);
```

```
end
```

```
if (flag ~= 1)
```

```
car(j,1) = car(j,1) + vAvg;
```

```
end
```

```
end
```

```
if ((car(j,1) > 700)&&(car(j,4) == NOTPASSED))
```

```
car(j,4) = PASSED;
```

```
carAmount = carAmount+1;
```

```
end
```

```
end
```

```
end
```

```
carAmount
```