

基于驾驶员行为分析的交通信号灯黄灯时长优化研究

顾晨曦 Chenxi.Gu22@student.xjtlu.edu.cn

范詠然 Heran.Fan22@student.xjtlu.edu.cn

须一菲 Yifei.Xu2202@student.xjtlu.edu.cn

摘要

本模型核心在于对黄灯相位周期内车辆行进过程的分段研究，并对合理黄灯时长的影响因子的作用进行具体探究，针对现行黄灯时长提出合理建议，针对问题一，我们采用控制变量方式，基于理想条件和现实参数分别建立匀速前进的模型一和变速的模型二，并对不同初速度机动车辆的不同情况进行分类。针对问题二，我们带入基于现实情况的各项参数数值，对计算得出的具体黄灯时长进行分析与现行较为普遍的 $3-4s$ 进行对比分析。

对于问题一，首先，我们构建了基于理想化交通环境的黄灯时长数学模型；其次，我们考虑了实际交通条件下的参数变化，如驾驶员反应时间、不同车型和道路条件，进一步建立了模型二；同时对不同初速度机动车辆的不同情况进行分类，分别探究初速度和绿灯变黄灯时距停止线距离对合理黄灯时长的具体影响。

对于问题二，利用控制变量法，分别探究有无黄灯倒计时，初速度大小，离停止线距离大小，行驶方向对于合理黄灯时长的影响，结果表明，在有黄灯倒计时，初速度更大，离停止线更远，直行情况下，为保证所有车辆都顺利通过交叉路口，应设置更久的黄灯时长来确保安全性。

对于问题三，为评估模型下机动车通过交叉路口的效率，本研究计算了车辆通过本侧停止线与对侧停止线所需时间的相关系数。通过两两间相关系数的计算，我们能够量化车辆通过交叉口的相互依赖性。在此基础上进一步绘制了热力图以直观展示不同车辆间的时间关联性，验证模型的可行性和优越性，并进一步分析模型潜在优势，提出改进方向。

结果表明，通过合理调整黄灯时长，可以显著提高车辆的通过率，减少交叉口内的交通事故。此外，我们还探讨了延长黄灯时长可能带来的潜在问题，如通行时间的不确定性增加和对交通流量的影响。

关键字： 边界建模 左转相位 Pearson 相关系数 二次多项式拟合

目录

一、 问题重述	3
1.1 问题背景	3
1.2 问题要求	3
二、 问题分析	4
2.1 问题一的分析	4
2.2 问题二的分析	4
2.3 问题三的分析	4
三、 模型假设	4
3.1 基本假设	4
3.2 模型一假设	5
四、 符号说明	5
4.1 问题一符号说明	5
五、 模型的建立与问题的求解	6
5.1 问题一的求解	6
5.1.1 模型一的建立	6
5.1.2 模型二的建立	6
5.2 问题二的求解	8
5.2.1 数值选取与分析	8
5.2.2 模型一的求解	9
5.2.3 模型二的求解	9
5.3 问题三的求解	12
六、 模型的评价，改进与推广	15
6.1 模型的评价	15
6.1.1 优点	15
6.1.2 缺点	15
6.2 模型的改进与推广	15
七、 模型的检验	16
7.1 相关性检验	16
参考文献	17

一、问题重述

1.1 问题背景

红绿灯是一种交通管制设施，通过红、黄、绿三种颜色的灯光交替变换，来规定道路通行权，确保车辆和行人的安全通行。交叉口是红绿灯的最常见使用场景，是城市道路的关键节点，更是交通事故的频发地。据统计，黄灯时间内发生的交通事故约占整个信号交叉口交通事故的 50% 以上。

为确保道路交通安全，我国已经出台多项强制性法规，明确规定了机动车在遇到黄灯信号时应遵循的行驶规则，以减少各类交通事故的发生。2004 年发布的《中华人民共和国道路交通安全法实施条例》第三十八条规定：“黄灯亮时，已越过停止线的车辆可以继续通行”，《道路交通安全法》规定黄灯表示警示，并规定机动车遇路口时应减速通过“黄灯亮时已经越过停止线的车辆可以继续通过，还未越过停止线的车辆应停车。抢黄灯行为属于违反道路交通信号灯通行，对驾驶人处 20 元以上 200 元以下罚款，记 6 分。”

然而，现实生活中，往往存在着车辆已经预备进入交叉路口但尚未完全通过或汽车距离停止线过近即使以最大加速度刹车也无法停止，但信号灯即将或已经从绿灯变为黄灯情况。这种情况下汽车可能无法及时制动，从而造成交通事故。通过研究相关影响因素，制定更加合理的黄灯时长，使正行驶在路口或离路口太近而无法安全停下的车辆能安全通过路口，可以有效降低此类交通事故发生可能性。

1.2 问题要求

1. 问题一：问题一为开放式问题，需要在未知情况下综合考虑初速度、行驶距离等各种与黄灯最优时长有关的直接、间接因素，模拟机动车在交通信号灯由绿转黄时汽车的行驶过程，分析归纳，建立合适的数学模型。
2. 问题二：在实际测算中，不同参数有相应取值范围，根据参数具体值我们可以确定黄灯的具体时长，以确保在不同的交通条件下驾驶员都能获得足够的反应时间来安全地停车或通过交叉口。综合考虑各个参数的具体值，模拟在不同路况下黄灯的合理时长。
3. 问题三：将模型运用到现实场景，将计算结果与现行标准进行比较，呈现模型优势与不足。基于此对比分析，提出针对性模型优化建议，以期提升模型的实用性和准确性。

二、问题分析

2.1 问题一的分析

在理想化的交通模型中，车辆从信号灯绿转黄至完全通过路口速度不变，需要考虑从绿灯变黄灯起至汽车车身完全通过路口，全程保持初始速度 v_0 的情况。

依据《中华人民共和国道路交通安全法》第四十四条规定：机动车通过交叉路口，应当按照交通信号灯、交通标志、交通标线或者交通警察的指挥通过；通过没有交通信号灯、交通标志、交通标线或者交通警察指挥的交叉路口时，应当减速慢行，且在通过路口时速度应小于 $30km/h$ 。因此需要进一步考虑当绿灯变黄灯时刻，司机经过反应踏减速踏板，进行减速过程下黄灯设置的不同时长。由于机动车的不同初速度，存在需要减速的不同距离，依据是否在停止线前将速度减至规定 $30km/h$ ，将模型分类。

2.2 问题二的分析

问题二立足于问题一模型，对过往文献中记录的驾驶员反应时间，不同道路上不同限速等车辆行为和路况数据进行了研究综述。在此基础上，我们特别关注了交通信号灯变换期间，左转车辆与直行车辆在路口所需行驶距离的差异性，以及公交车与轿车等不同车型因其不同车身长度而对路口通行效率所产生的影响。

此外，考虑到黄灯闪烁时是否有装配倒计时装置是影响整个交通流程总时长的关键因素，本研究进一步探讨了这一因素对合理设置黄灯时长的具体影响。

为了系统地研究这些变量之间的关系，需采用了控制变量法，对模型一和模型二在不同条件下的适用性进行对比分析，考察某一变量变化对黄灯时长设置的潜在影响。

2.3 问题三的分析

考虑到模型应用于实际场景时所面临的复杂可能性，将第二问得出的黄灯时长优化结果与现行的黄灯时长标准进行综合比较分析，在过程中进一步考虑此前未曾涉及的可能性。同时进行对比分析，分别计算在现行黄灯时长和模型模拟黄灯时长下末行车辆分别对与本侧停止线和对侧停止线通过的可能性，分析此指标优劣以及相关原因，进一步说明模型的潜在优越性。

三、模型假设

3.1 基本假设

1. 本题仅探讨左转和直行车辆，假设右转车辆对其他车道车辆无影响。

2. 假设所研究区域所有车辆遵守《中华人民共和国道路交通安全法》，在路口内不进行加速。
3. 假设司机在看到绿灯变黄灯时反应和决策过程中汽车保持匀速前进。
4. 假设分析对象前方没有车辆，即对象处于开阔的道路条件下，其他车辆对分析对象的速度无影响。
5. 假设不存在侧方突然冲出的行人，非机动车等不确定因素。
6. 假设模型中所涉及的函数都是随时间连续变化的，可微的函数。

3.2 模型一假设

1. 假设红绿灯在此种情况黄灯变红灯时无倒计时。
2. 假设左转和直行车辆在看到绿灯变黄时均不减速。

四、符号说明

4.1 问题一符号说明

序号	符号	符号说明
1	t_0	驾驶员反应和决策时间 (s)
2	x_0	驾驶员反应和决策时间内行驶距离 (m)
3	t_1	车辆减速时间 (s)
4	t_2	驾驶员做出决策至车头触碰停止线所经历的时间 (s)
5	L	车身长度 (m)
6	H	路口宽度 (m)
7	v_0	$t = 0$ 汽车初始速度 (m/s)
8	v_F	法规规定的通过路口的标准速度 (30m/s)(m/s)
9	a_1	$t = 0$ 在保证安全的情况下汽车减速最大制动力下的加速度 (m/s)
10	a_2	$t = 0$ 参考减速加速度 (m/s)
11	a_0	$t = 0$ 参考减速加速度 (m/s)
12	S	绿灯变黄灯时距停止线距离 (m)

五、模型的建立与问题的求解

5.1 问题一的求解

5.1.1 模型一的建立

依据现实情况将驾驶员从反应到通过路口过程依据驾驶员反应时间分为三段，及驾驶员反应和决策时间内汽车行驶过程 x_0 ，驾驶员采取行动至车头触碰停止线形式距离 x_1 ，汽车在保证安全的情况下最大减速度为 a_1 ，路口长度为 H ，车身长度为 L 。

汽车行驶离路口太近而无法安全停下的车辆满足：

$$v_0 + a_1 t_1 > 0$$

选取正好离路口太近而无法安全停下的末行车辆，即为确保所有车辆都能安全通过路口，黄灯亮起的时间 T 满足：

$$T = t_0 + \frac{\int (v_0 + a_1 t_2) dt + L + h}{v_0}$$

即

$$T = \frac{S + L + H}{v_0}$$

5.1.2 模型二的建立

根据不同减速过程将情况分为五类，如图1所示：

- 驾驶员在绿灯变黄灯瞬间猛踩刹车（保持最大加速度减速），车尾触碰对侧路口停止线时刻速度依然大于 $30km/h$
- 驾驶员在绿灯变黄灯瞬间猛踩刹车（保持最大加速度减速），在路口中速度可以减至 $30km/h$ ，后续通过路口过程匀速
- 驾驶员在绿灯变黄灯瞬间猛踩刹车（保持最大加速度减速），在车头触碰停止线前（时）速度可以减至 $30km/h$ ，通过整段路口过程匀速
- 驾驶员在绿灯变黄灯瞬间猛踩刹车（保持最大加速度减速），在车头触碰停止线前速度可以减至 $0km/h$
- 驾驶员在绿灯变黄灯瞬间轻踩刹车，在车头触碰停止线时速度可以减至 $0km/h$

由于后两种情况中，驾驶员能够在汽车抵达停止线之前成功将车辆停稳，因此接下来的分析将仅针对前三种情形进行深入探讨。

1. 情况一：汽车全程以 a_1 进行匀减速

$$\begin{cases} v_{end}^2 - v_0^2 = 2a_1 x \\ x = S + L + H - v_0 t_0 \\ v_{end} - v_0 = a_1 t_1 \end{cases}$$

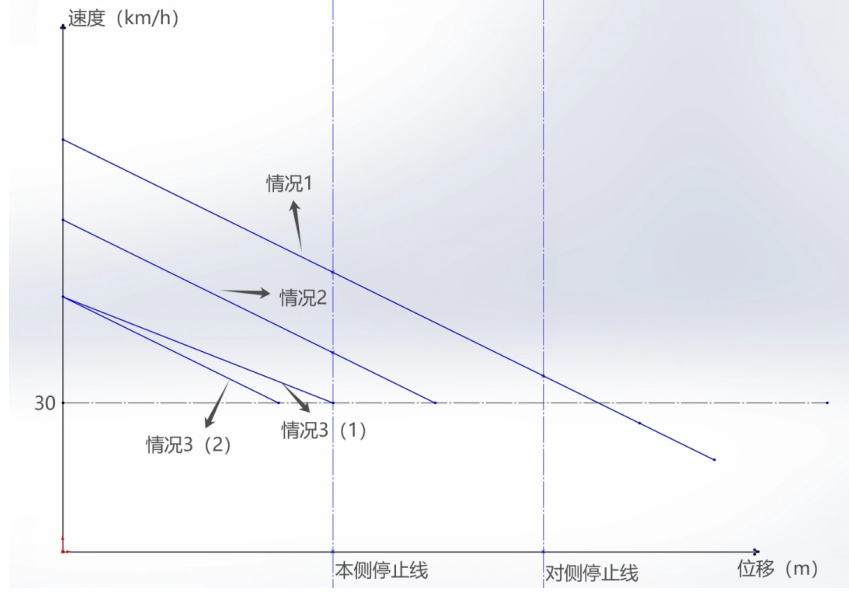


图 1 模型二分类示意图

即

$$T = \frac{\sqrt{v_0^2 + 2a_1(S + L + H - v_0 t_0)} - v_0}{a_1} + t_0$$

2. 情况二：汽车全程先以 a_1 进行匀减速，再以 $30km/h$ 匀速通过路口

$$\begin{cases} t_1 = \frac{v_F - v_0}{a_1} \\ s + x_1 - v_0 t_0 = \frac{v_F^2 - v_0^2}{2a_1} \\ x_2 = H + L - x_1 \\ x_2 = t_2 v_F \end{cases}$$

即

$$T = \frac{v_F - v_0}{a_1} + \frac{H + L - \frac{v_F^2 - v_0^2}{2a_1} - v_0 t_0 + S}{v_F} + t_0$$

3. 情况三（1）：汽车在到达停止线时刹至 $30km/h$ ，剩余路程匀速

$$\begin{cases} a_2 = \frac{v_F^2 - v_0^2}{2(S - v_0 t_0)} \\ a_2 t_1 = v_F - v_0 \\ t_2 v_F = H + L \end{cases}$$

即

$$T = \frac{2(S - v_0 t_0)}{v_F + v_0} + \frac{H + L}{v_F} + t_0$$

4. 情况三（2）：汽车在到达停止线前刹至 $30km/h$ ，剩余路程匀速

$$\begin{cases} a_1 t_1 = v_F - v_0 \\ 2a_1 x_1 = v_F^2 - v_0^2 \\ v_F t_2 = H + L + S - \frac{v_F^2 - v_0^2}{2a_1} - v_0 t_0 \end{cases}$$

即

$$T = \frac{v_F - v_0}{a_1} + \frac{H + L + S - \frac{v_F^2 - v_0^2}{2a_1} - v_0 t_0}{v_F} + t_0$$

5.2 问题二的求解

为进行黄灯合理时长的确定，需要探究司机的反应感知时间（ t_0 ）和最大减速度（ a_1 ），其中，反应感知时间的确定需要数据的收集以确定其对应函数模型，其影响因素主要为初速度和黄灯是否有倒计时。

5.2.1 数值选取与分析

考虑在某些特定区域存在在黄灯转红灯有倒计时情况，探究有无倒计时对汽车安全通过时间，即最优黄灯亮灯时长的影响。有倒计时情况下，驾驶员看见倒计时即以更大加速度开始减速，加速度的选取尤为重要。

• a_1 和 t_0 的选取

- 根据 [1]，在无读秒条件下，通过拟合，感知反应时间与速度之间的拟合结果为表2a，指数函数较其他备选函数，能更好地模拟倒计时条件下最大减速度与黄灯启亮时速度之间的变化关系，公式为：

$$a_1 = -\exp(2340 - \frac{52.329}{3.6v_0})$$

- 在有读秒条件下，感知反应时间与速度之间的拟合结果为表2b，对数函数较其他备选函数，能更好地模拟倒计时条件下最大减速度与黄灯启亮时速度之间的变化关系，公式为：

$$a_1 = 0.045v_0^{1.087}$$

- 在无读秒条件下，感知反应时间与速度之间的拟合结果为表2c，指数函数为最合适的函数模型，公式如下：

$$t_0 = 2.5\exp(-0.018) * 3.6v_0$$

- 在有读秒条件下，感知反应时间与速度之间的拟合结果为表2d，对数函数为最合适的函数模型，公式为：

$$t_0 = -1.926 \ln(3.6v_0) + 8.703$$

函数名称	R^2 值	F 值	样本大小	p 值	常数	系数
线性	0.513	33.769	34	<0.001	2.147	-0.020
对数	0.551	39.263	34	<0.001	4.894	-0.987
倒数	0.474	28.821	34	<0.001	0.310	33.313
幂	0.612	50.429	34	<0.001	26.345	-0.858
S 型	0.498	31.709	34	<0.001	-0.697	28.184
指数	0.615	51.013	34	<0.001	2.500	-0.018

(a) 无倒计时条件下最小感知反应时间与黄灯启亮时速度之间关系的拟合结果

函数名称	R^2 值	F 值	样本大小	p 值	常数	系数
线性	0.822	147.743	34	<0.001	0.113	0.066
对数	0.896	274.300	34	<0.001	-8.861	3.213
倒数	0.830	155.852	34	<0.001	6.169	-112.616
幂	0.847	177.033	34	<0.001	0.015	1.376
S 型	0.924	389.202	34	<0.001	2.340	-52.329
指数	0.654	60.603	34	<0.001	0.793	0.026

(c) 无倒计时条件下最大减速度与黄灯启亮时速度之间关系的拟合结果

函数名称	R^2 值	F 值	样本大小	p 值	常数	系数
线性	0.721	146.976	59	<0.001	3.243	-0.040
对数	0.779	201.289	59	<0.001	8.703	-1.926
倒数	0.764	184.275	59	<0.001	-0.448	78.745
幂	0.726	151.284	59	<0.001	390.873	-1.515
S 型	0.666	113.516	59	<0.001	-1.180	59.886
指数	0.717	144.492	59	<0.001	5.596	-0.032

(b) 有倒计时条件下最小感知反应时间与黄灯启亮时速度之间关系的拟合结果

函数名称	R^2 值	F 值	样本大小	p 值	常数	系数
线性	0.809	241.048	59	<0.001	-0.348	0.072
对数	0.766	186.938	59	<0.001	-9.339	3.246
倒数	0.665	113.381	59	<0.001	5.899	-124.909
幂	0.812	245.566	59	<0.001	0.045	1.087
S 型	0.749	170.522	59	<0.001	2.041	-43.125
指数	0.805	235.149	59	<0.001	0.954	0.023

(d) 有倒计时条件下最大减速度与黄灯启亮时速度之间关系的拟合结果

图 2 有无倒计时情况下参数函数拟合结果

	有倒计时	无倒计时
$a_1(v_0)$	$a_1 = 0.045v_0^{1.087}$	$a_1 = -exp(2340 - \frac{52.329}{3.6v_0})$
$t_0(v_0)$	$t_0 = -1.926 \ln(3.6v_0) + 8.703$	$t_0 = 2.5exp(-0.018) * 3.6v_0$

表 1 参考数据公式汇总

5.2.2 模型一的求解

根据数值选取中一般情况，取 $v_0 = 15m/s$, $S = 10m$, $L = 4m$, $H = 30m$ ，得到

$$T = \frac{S + L + H}{v_0} \approx 3s$$

结果约等于现有常见的黄灯时长，利用模型二进一步探究各个因素对黄灯时长的具体影响。

5.2.3 模型二的求解

- S 范围的选取根据文献 [2] 的调研结果，研究发现，无论在信号灯黄灯阶段结束前的倒计时显示与否，当车辆距离交叉路口的距离 $S < 20m$ ，几乎所有驾驶员倾向于继续行驶而不采取停车措施。相对地，当距离 $S > 60m$ 时，绝大多数驾驶员则选择开始减速并准备停车。基于此，黄灯时长参数的取值范围应为 $20m - 60m$ ，更准确地模拟驾驶员在不同距离下的决策行为，在此基础上探究有无黄灯结束倒计时，变量 S 和变量 v_0 对黄灯合理时间 T 的影响。根据图3, $20 - 40m$ 为司机犹豫和决策的主要区域，因此本研究最终选取 $20 - 40m$ 作为 S 取值范围。

1. 有无黄灯结束倒计时对汽车安全通过时间 T 的影响

- 在固定安全时间 S 的条件下，如图4所示，随着初始速度 v_0 的增加，所需黄灯时

	限速值 (km/h)		
	无道路中心线	同方向单车道	同方向多车道
城市道路	30	50	70
公路	40	70	80

表 2 中国城市道路类型及其限速值

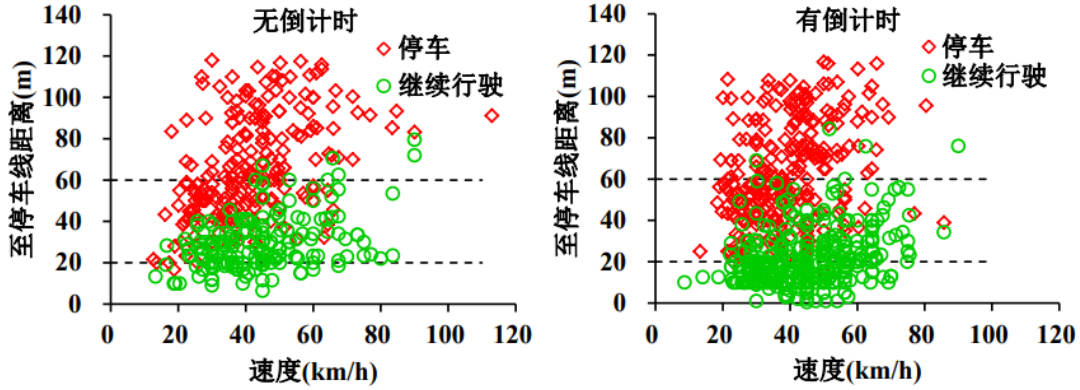


图 3 在无倒计时（左图）和有倒计时（右图）条件下驾驶行为 [1]

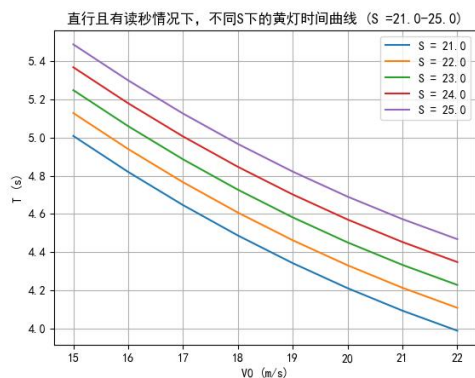
间 T 呈现减少的趋势。特别地，在无黄灯倒计时的情况下，当初始速度为取值区间最小值 $v_0 = 15m/s$ ，对应的黄灯时间 T 相对较大。同时，在无倒计时情况下当其他变量保持不变，随着初始速度的增加，合理设置的黄灯时间 T 下降速率更为显著。固定 S ，如4所示，随 v_0 增加 T 减小，当变量相同，在有倒计时情况下，随 v_0 增大所需合理黄灯时间 T 下降更快。

- 固定 v_0 ，如图6，随距离 S 的逐渐增加，所需黄灯时间 T 呈现增加的趋势。在无黄灯倒计时的情况下，当初始速度为取值区间最小值 $v_0 = 15m/s$ ，对应的黄灯时间 T 相对较大。同时，在无倒计时情况下当其他变量保持不变，随着距离 S 的增加，合理设置的黄灯时间 T 上升更为显著。

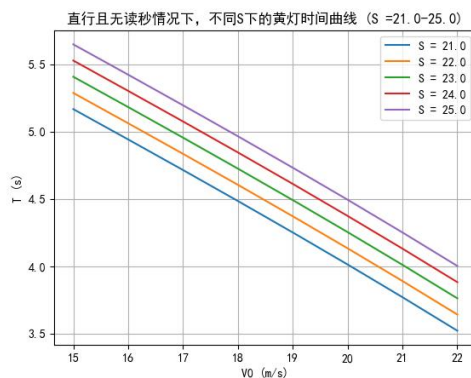
2. S 对于汽车安全通过时间 T 的影响

- 当 S 固定，图像4和5显示了在不同绿灯变黄灯时距停止线距离 S 数值下合理黄灯时间 T 随 v_0 变化的曲线， S 的取值范围为 $[21, 40]$ （单位为 m ）。随着初速度的增加，曲线呈现下降趋势且斜率随之减小。这意味着在给定 S 的条件下，速度越快，完成整个过程所需的时间越少。
- 当 v_0 固定，图像6显示了在不同 v_0 下合理黄灯时间 T 随 v_0 变化的曲线， S 的取值范围为 $21m - 30m$ 。随着 S 的增加，合理黄灯时长 T 曲线整体增加。

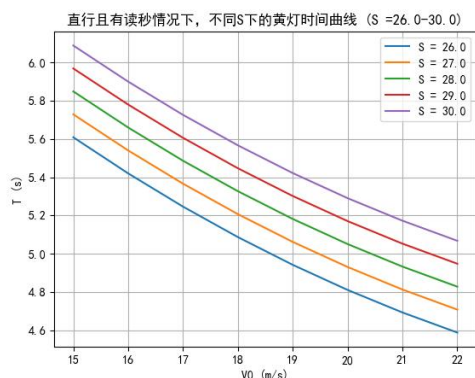
3. v_0 对于汽车安全通过时间 T 的影响



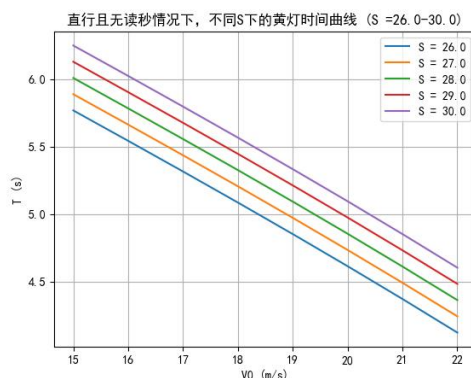
(a) $S \in [21, 25]$, 有倒计时



(b) $S \in [21, 25]$, 无倒计时



(c) $S \in [26, 30]$, 有倒计时



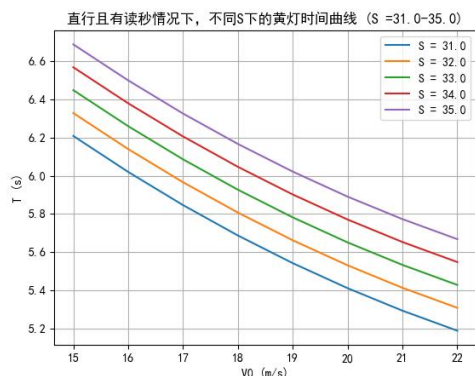
(d) $S \in [26, 30]$, 无倒计时

图 4 不同 S 值下的时间曲线比较 (1)

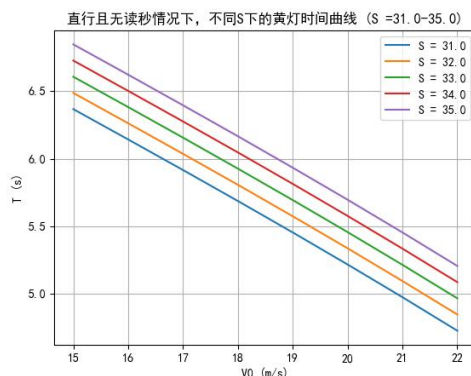
- 当 S 固定, 图像5显示了在不同 S 下合理黄灯时间 T 随 v_0 变化的曲线, v_0 的取值范围为 $15\text{m/s} - 22\text{m/s}$, 即 $54\text{km/h} - 79.2\text{km/h}$ 。随着初速度的增加, 曲线呈现下降趋势且斜率随之减小。这意味着在给定 S 的条件下, 速度越快, 完成整个过程所需的时间越少。
- 当 v_0 固定, 图像6显示了在不同 v_0 下合理黄灯时间 T 随 v_0 变化的曲线, S 的取值范围为 $21\text{m} - 30\text{m}$ 。随着 S 的增加, 合理黄灯时长 T 曲线整体增加, 且 v_0 越小, T 增加越快。

4. 不同车道对于汽车安全通过时间 T 的影响

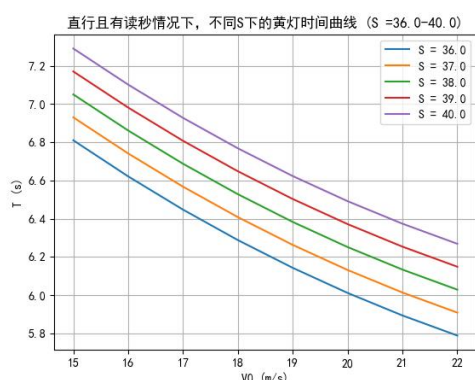
- 如图7所示, 在路口宽度为 H 情况下左转车道正常车辆行驶距离约为 $\frac{7}{24}\pi H$, 将其带入模型二, 分别比较在固定 S 和 v_0 时直行和左转时合理黄灯时长 T 的不同。
- 图像8a和8b显示了在不同 S 下合理黄灯时间 T 随 v_0 变化的曲线, 图像8c和8d显示了在不同 v_0 下合理黄灯时间 T 随 S 变化的曲线, 观察图像可以发现, 直行需要的黄灯时长普遍高于相同情况下左转所需时长, 在现实生活运用中可以据此为左转道和直行道设置不同的黄灯长度, 节省时间成本提高效率。



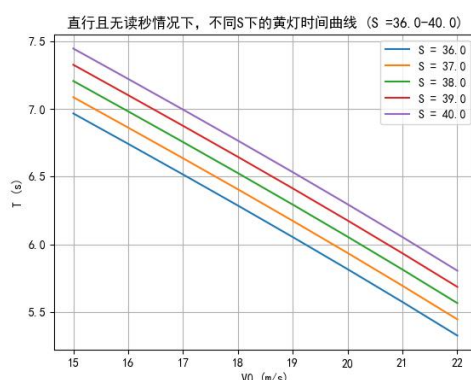
(a) $S \in [31, 35]$, 有倒计时



(b) $S \in [31, 35]$, 无倒计时



(c) $S \in [36, 40]$, 有倒计时



(d) $S \in [36, 40]$, 无倒计时

图 5 不同安全时间 S 值下的时间曲线比较 (2)

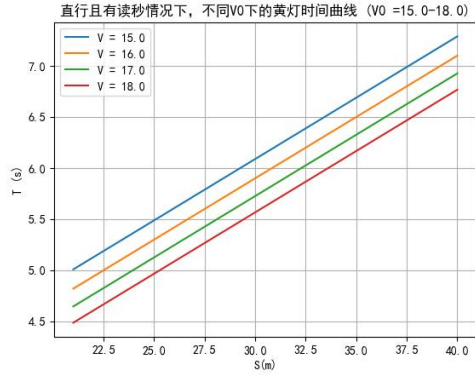
5.3 问题三的求解

问题二解得的合理黄灯时长（得出的黄灯时长主要由末行车的行驶时间决定）大致在 $6-7s$, 高于现行得到一般标准 $3-4s$, 大致原因如下:

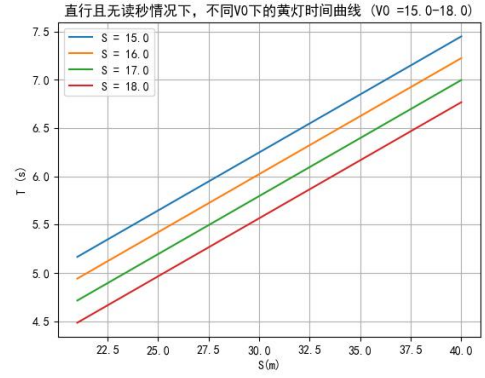
1. 如表3, 在现行黄灯执行标准 $3-4s$ 内, 大部分车辆都能安全通过本侧停止线, 进入交叉路口, 但仅有少部分车辆（当黄灯时长为 $3s$ 时甚至没有车能通过交叉路口）能通过对侧停止线, 大部分车辆在黄灯转红时依然逗留在交叉路口内, 安全隐患较大。但当将标准提高至上述模型计算值时, 基本所有车辆都能安全通过交叉路口。
2. 在计算中考虑了为了严格遵守交规所进行的适当减速, 因此会增加末行车的总体行驶时间, 从而造成合理黄灯时间 T 的增加
3. 在计算中考虑了防止碰撞行人和前方减速车辆等突发情况而进行的适当减速, 一定在程度上也增加了合理黄灯时间
4. 在计算中考虑了每辆车的车身长度, 使末行车行驶时间变长

黄灯市场较长带来的潜在优势有:

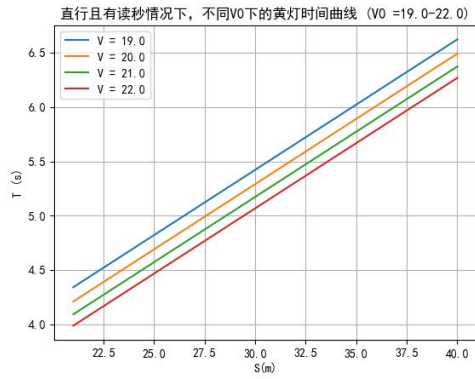
- 提供充分的反应时间: 延长的黄灯时长为驾驶员提供了额外的决策时间, 尤其在交



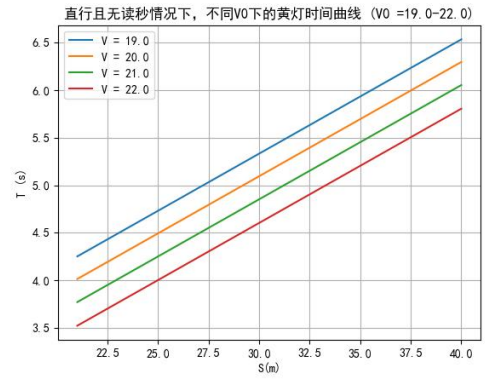
(a) $v_0 \in [15, 18]$, 无倒计时



(b) $v_0 \in [15, 18]$, 无倒计时



(c) $v_0 \in [19, 22]$, $S = 40$, 无倒计时



(d) $v_0 \in [19, 22]$, 无倒计时

图 6 不同初始速度和安全时间下的时间曲线比较

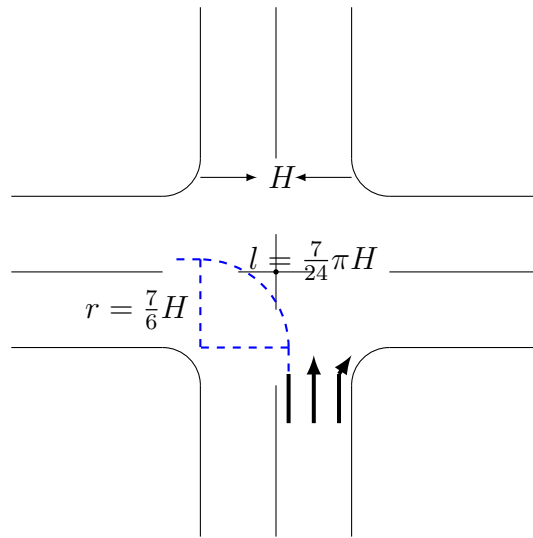
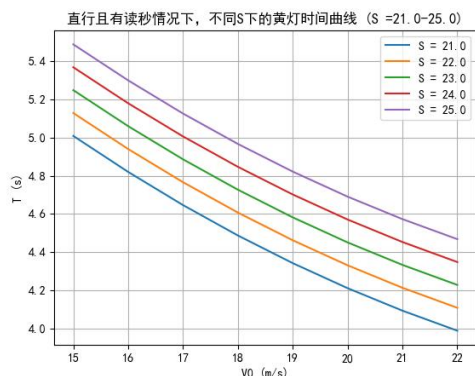
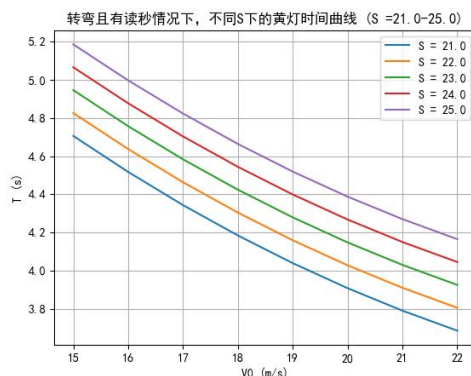


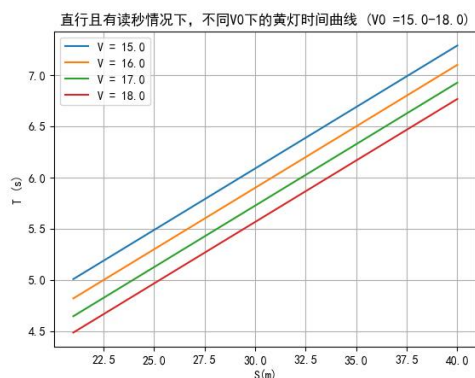
图 7 左转道与直行道路口经过距离 H



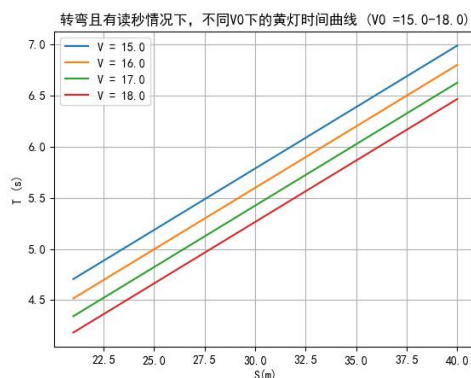
(a) S 固定, 直行有倒计时



(b) S 固定, 左转有倒计时



(c) v_0 固定, 直行有倒计时



(d) v_0 固定, 左转有倒计时

图 8 不同方向下的合理黄灯时间曲线比较

通高峰或不利天气条件下显得尤为重要。

- 减少急刹车情况：充足的黄灯时长允许驾驶员有更多的时间来做出反应，从而可能降低因紧急制动而导致的交通事故发生概率。
- 增强道路利用率和流畅性：较长的黄灯时长有助于更多车辆在信号灯变换前通过路口，有助于缓解交通拥堵，提高道路使用率和效率。

黄灯时长较长也存在一些潜在的缺点：

- 增加通行时间的不确定性：过长的黄灯时长可能使驾驶员难以准确决策是否、何时应加速通过路口，这种不确定性可能对交通反而产生负面影响。
- 对路口容量的影响：在交通密集区域，较长的黄灯时长可能导致路口通行能力降低，从而引发交通拥堵和延误。
- “冒险主义者”：黄灯时长的延长可能诱使部分驾驶员在黄灯期间冒险通过，反而增强了违规的风险。
- 在当前的研究中，尚未计入两侧车道交通信号灯由红色转为绿色时车辆的起步时间以及驾驶员的反应时间。这一忽略可能一定程度上导致计算所得的合理黄灯时长结果偏高，从而造成时间资源的浪费。

黄灯时长 (s)	通过停止线车辆百分比 (%)	对侧停止线百分比 (%)
3	88.44%	0.00%
4	100.00%	2.19%
5	100.00%	27.81%
6	100.00%	67.81%
7	100.00%	96.88%
8	100.00%	100.00%

表 3 黄灯时长与通过停止线车辆百分比的关系

六、模型的评价，改进与推广

6.1 模型的评价

6.1.1 优点

- 该模型设计考虑了广泛的适用性，能够适应多变的交通环境和驾驶员行为。
- 模型对不同交通情境进行了细致的分类，并针对每一类别进行了深入的讨论，以确保模型的有效性，通过个性化定制，黄灯时间的设定展现出更高的灵活性，从而优化交通流的控制和提高道路安全。
- 在本研究中，所选取的加速度值并非最大理论值，而是在确保行车安全的前提下，经过审慎考量后确定的合理数值，旨在实现在不同交通条件下的最优驾驶行为。

6.1.2 缺点

- 该模型建立在严格遵守交通规则假设之上，但在现实生活中通常存在着在黄灯的最后几秒驾驶员“冲线”加速的情况，此类情况下加速度为正，需另行考虑。
- 模型需要进一步考虑现实的偶发因素，如极端天气，路面情况 [3] 等，这些因素可能对驾驶员的反应时间和初速度产生显著影响。

6.2 模型的改进与推广

- 在本次解题中我们未对各影响因子进行相关性分析，在收集到更完备的数据的前提下，我们可以构建聚类模型，将具有相似性质的因素进行归类，求出关于黄灯信号时长 T 更完备的方程
- 在有利润方程的前提下，设置一定约束条件，可以采用线性规划模型进行运算得到更准确的结果。

- 在现实生活中还有可能存在绿灯变黄灯存在倒计时的情况，需要进一步讨论。

七、模型的检验

7.1 相关性检验

利用问题三得到数据，我们对模型涉及参数进行相关性检验，皮尔森相关系数热力图结果如图9所示：

研究结果表明，车辆在本侧停止线末尾的通过百分比以及在对侧停止线末尾的通过百分比与黄灯时长之间存在显著的线性相关性。进一步地，通过对车辆通过率（即车辆尾部已完全通过对侧停止线）与黄灯时长进行线性回归分析，我们得到了以下回归方程：

$$\text{通过率} = -0.01584821428571414x^2 + 0.40808035714285573x - 1.174017857142853$$

其中， x 代表黄灯时长。根据该方程可以推断，在当前黄灯时长的基础上，适当延长黄灯时间将有助于提高车辆的通过率。我们可以通过提高黄灯时间提高汽车通过率，提升城市交通车流安全性。

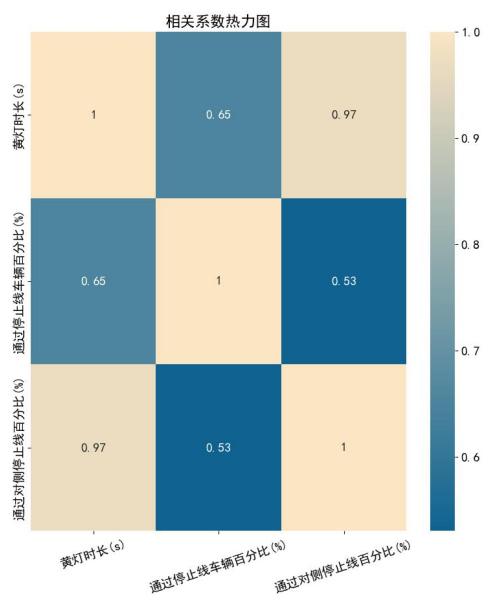


图9 相关系数热力图矩阵

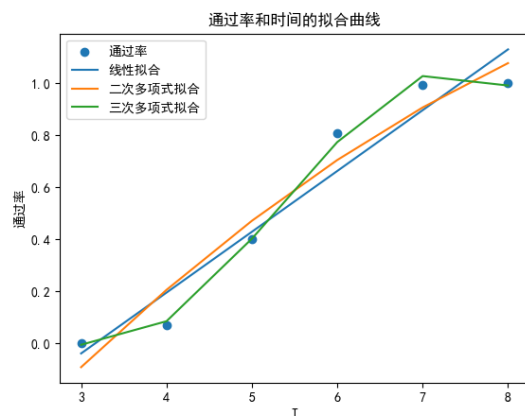


图 10 通过时率与黄灯时间拟合

参考文献

- [1] 付云川. 信号交叉口黄灯两难区边界建模及风险评估方法研究 [A]. 高等教育出版社, 2015: 12.
- [2] 李克平, 杨佩昆, 倪颖. 城市道路交叉口信号控制中的黄灯问题 [J]. 城市交通, 2010, 8(4): 67-72
- [3] 焦兴旺, 张福生, 黄柄胜, 赵寅. 基于黄灯困境区的相位安全切换时机研究 [J]. 湖南交通科技, 2023, 49(4): 184-191.

程序一：Python 模拟模型二合理黄灯时长：

```
1 import os
2 import math
3 import numpy as np
4 import pandas as pd
5 import matplotlib.pyplot as plt
6
7 save_path = r'D:\Mathematical_Modeling\training\pics'
8 path1 = r'D:\Mathematical_Modeling\training\rate.csv'
9
10 if not os.path.exists(save_path):
11     os.makedirs(save_path)
12
13 plt.rcParams['font.sans-serif'] = ['SimHei']
14 plt.rcParams['axes.unicode_minus'] = False
15
16 CT=[0,1] # Countdown or not
17 L=4 # lenth of cars
18 Hwan=30*3.14*7/24 # Intersection width
19 Hzhi=30
20 NT2=0;NT31=0;NT32=0 # counters
21 S_array=np.linspace(21,40,20) # Distance to the stop line
22 V0_array=np.linspace(15,22,8) # Initial Velocity
23 T_counter=[] # us
24 t_counter=[] # normal
25
26
27
28
29 def a1t0withCT(V):
30     a1=-0.045*(3.6*V)**1.087
31     t0=-1.926*math.log(3.6*V)+8.703
32     return a1,t0
33
34 # without Countdown
```

```

35 def a1t0withoutCT(V):
36     a1=-math.exp(2.340-52.329/(3.6*V))
37     t0=2.5*math.exp(-0.018*3.6*V)
38     return a1,t0
39
40
41 def T_time(V0,VF,a1,t0,S,L,H):
42     judge1=V0*t0-V0**2/(2*a1)
43
44     judge2=V0*t0+(VF**2-V0**2)/(2*a1)
45     # print(judge1,judge2)
46     if (S>judge1):
47         print('寄')
48         return (float('nan'),float('nan'),float('nan'))
49     if(S<=judge1):
50         if (judge2>S+L+H):
51             T1=(V0-(V0**2+2*a1*(S+L+H-V0*t0))**0.5)/(-a1)+t0
52             t1_pass=(-V0+(V0**2-2*a1*(V0*t0-S-L))**0.5)/a1+t0
53             # print(T1,V0,S,'1')
54             return(T1,t1_pass)
55
56         if(S<judge2<=S+L+H):
57             T2=(VF-V0)/a1 + (H+L-((VF**2-V0**2)/(2*a1))+S-V0*
t0)/VF + t0
58             t2_pass=(-V0+(V0**2-2*a1*(V0*t0-S-L))**0.5)/a1+t0
59             # print(T2,V0,S,'2')
60             return(T2,t2_pass)
61
62         if(S>judge2>0):
63             T31=(VF-V0)/a1+(H+L+S-V0*t0-(VF**2-V0**2)/(2*a1))/
VF+t0
64             t31_pass=2*(S-V0*t0)/(VF+V0)+L/VF+t0
65             # print(T31,V0,S,'3')
66             return(T31,t31_pass)
67

```

```

68
69 def bianli(CT,S_array,V0_array):
70     T_final=[]
71     t_final=[]
72
73     for CT1 in CT:
74         n=0
75         for S in S_array:
76             T_array = []
77             t_array = []
78             n=n+1
79             for V0 in V0_array:
80                 if CT1:
81                     a1,t0=a1t0withCT(V0)
82                 else:
83                     a1,t0=a1t0withoutCT(V0)
84                 T,t=T_time(V0,8.33,a1,t0,S,L,Hwan)
85                 T_final.append(T)
86                 t_final.append(t)
87                 T_array.append(T)
88                 t_array.append(t)
89                 # print(T,t,S,V0)
90             if CT1==1:
91                 # print('1')
92                 plt.xlabel('V0 (m/s)')
93                 plt.ylabel('T (s)')
94                 plt.plot(V0_array, T_array, label=f'S = {S}')
95                 if n==5:
96                     plt.title(f'直行且有读秒情况下，不同S下的
黄灯时间曲线 (S = {S-4}-{S})')
97                     plt.legend()
98                     plt.grid(True)
99                     # plt.show()
100                    plt.savefig(f'{save_path}/
Time_Curve_S_WCT_Turn{S:.2f}.jpg')

```

```

101         plt.close()
102         n=0
103     if CT1==0:
104         # print('1')
105         plt.xlabel('V0 (m/s)')
106         plt.ylabel('T (s)')
107         plt.plot(V0_array, T_array, label=f'S = {S}')
108         if n==5:
109             plt.title(f'直行且无读秒情况下，不同S下的
黄灯时间曲线 (S = {S-4}-{S})')
110             plt.legend()
111             plt.grid(True)
112             # plt.show()
113             plt.savefig(f'{save_path}/
Time_Curve_S_WOCT_Turn{S:.2f}.jpg')
114             plt.close()
115             n=0
116     for V0 in V0_array:
117         T_array = []
118         n=n+1
119         for S in S_array:
120             if CT1:
121                 a1,t0=a1t0withCT(V0)
122             else:
123                 a1,t0=a1t0withoutCT(V0)
124
125             T,t=T_time(V0,8.33,a1,t0,S,L,Hwan)
126             T_final.append(T)
127             t_final.append(t)
128             T_array.append(T)
129             t_array.append(t)
130             # print(T,t,S,V0)
131     if CT1==1:
132         # print('1')
133         plt.xlabel('S(m)')

```

```

134         plt.ylabel('T (s)')
135         plt.plot(S_array, T_array, label=f'V = {V0}')
136         if n==4:
137             plt.title(f'直行且有读秒情况下，不同V0下的
黄灯时间曲线 (V0 ={V0-3}-{V0})')
138             plt.legend()
139             plt.grid(True)
140             # plt.show()
141             plt.savefig(f'{save_path}/
Time_Curve_V0_WCT_Turn{V0}.jpg')
142             plt.close()
143             n=0
144         if CT1==0:
145             # print('1')
146             plt.xlabel('S(m)')
147             plt.ylabel('T (s)')
148             plt.plot(S_array, T_array, label=f'S = {V0}')
149             if n==4:
150                 plt.title(f'直行且无读秒情况下，不同V0下的
黄灯时间曲线 (V0 ={V0-3}-{V0})')
151                 plt.legend()
152                 plt.grid(True)
153                 # plt.show()
154                 plt.savefig(f'{save_path}/
Time_Curve_V0_WOCT_Turn{V0}.jpg')
155                 plt.close()
156                 n=0
157
158
159     for CT1 in CT:
160         n=0
161         for S in S_array:
162             T_array = []
163             t_array = []
164             n=n+1

```

```

165         for V0 in V0_array:
166             if CT1:
167                 a1,t0=a1t0withCT(V0)
168             else:
169                 a1,t0=a1t0withoutCT(V0)
170             T,t=T_time(V0,8.33,a1,t0,S,L,Hwan)
171             T_final.append(T)
172             t_final.append(t)
173             T_array.append(T)
174             t_array.append(t)
175             # print(T,t,S,V0)
176         if CT1==1:
177             # print('1')
178             plt.xlabel('V0 (m/s)')
179             plt.ylabel('T (s)')
180             plt.plot(V0_array, T_array, label=f'S = {S}')
181             if n==5:
182                 plt.title(f'转弯且有读秒情况下，不同S下的
黄灯时间曲线 (S = {S-4}-{S})')
183                 plt.legend()
184                 plt.grid(True)
185                 # plt.show()
186                 plt.savefig(f'{save_path}/
Time_Curve_S_WCT_Turn{S:.2f}.jpg')
187                 plt.close()
188                 n=0
189         if CT1==0:
190             # print('1')
191             plt.xlabel('V0 (m/s)')
192             plt.ylabel('T (s)')
193             plt.plot(V0_array, T_array, label=f'S = {S}')
194             if n==5:
195                 plt.title(f'转弯且无读秒情况下，不同S下的
黄灯时间曲线 (S = {S-4}-{S})')
196                 plt.legend()

```

```

197         plt.grid(True)
198         # plt.show()
199         plt.savefig(f'{save_path}/
Time_Curve_S_WOCT_Turn{S:.2f}.jpg')
200         plt.close()
201         n=0
202     for V0 in V0_array:
203         T_array = []
204         n=n+1
205         for S in S_array:
206             if CT1:
207                 a1,t0=a1t0withCT(V0)
208             else:
209                 a1,t0=a1t0withoutCT(V0)
210
211             T,t=T_time(V0,8.33,a1,t0,S,L,Hwan)
212             T_final.append(T)
213             t_final.append(t)
214             T_array.append(T)
215             t_array.append(t)
216             # print(T,t,S,V0)
217         if CT1==1:
218             # print('1')
219             plt.xlabel('S(m)')
220             plt.ylabel('T (s)')
221             plt.plot(S_array, T_array, label=f'V = {V0}')
222             if n==4:
223                 plt.title(f'转弯且有读秒情况下，不同V0下的
黄灯时间曲线 (V0 ={V0-3}-{V0})')
224                 plt.legend()
225                 plt.grid(True)
226                 # plt.show()
227                 plt.savefig(f'{save_path}/
Time_Curve_V0_WCT_Turn{V0}.jpg')
228                 plt.close()

```



```

229         n=0
230     if CT1==0:
231         # print('1')
232         plt.xlabel('S(m)')
233         plt.ylabel('T (s)')
234         plt.plot(S_array, T_array, label=f'S = {V0}')
235         if n==4:
236             plt.title(f'转弯且无读秒情况下，不同V0下的
黄灯时间曲线 (V0 ={V0-3}-{V0})')
237             plt.legend()
238             plt.grid(True)
239             # plt.show()
240             plt.savefig(f'{save_path}/
Time_Curve_V0_WOCT_Turn{V0}.jpg')
241             plt.close()
242             n=0
243     return T_final,t_final
244
245
246
247 T_pass,t_pass=bianli(CT,S_array,V0_array)
248 # print(len(T_pass))
249 # print(len(t_pass))
250
251 T_rate_array=[]
252 t_rate_array=[]
253 for t in [3,4,5,6,7,8]:
254     T_count_greater_than_T = sum(1 for item in T_pass if item
< t)
255     t_count_greater_than_t = sum(1 for item in t_pass if item
< t)
256
257     print(f'T={t}时，通过率是{T_count_greater_than_T/len(
T_pass)}')
258     print(f't={t}时，通过率是{t_count_greater_than_t/len(

```

```

    t_pass))}')
259
260     T_rate_array.append(100*(T_count_greater_than_T/len(T_pass)
    ))
261     t_rate_array.append(100*(t_count_greater_than_t/len(t_pass)
    ))
262
263 df = pd.DataFrame({'黄灯时长(s)': [3,4,5,6,7,8], '通过对侧停止
    线百分比(%)': T_rate_array, '通过停止线车辆百分比(%)':
    t_rate_array })
264
265 df.to_csv(path1,index=False)
266
267 # t=3时，通过率是0.884375
268                                     # T=3时，通过率是0.0
269 # t=4时，通过率是1.0
270                                     # T=4时，通过率是0.06875
271 # t=5时，通过率是1.0
272                                     # T=5时，通过率是0.4
273 # t=6时，通过率是1.0
274                                     # T=6时，通过率是0.80625
275 # t=7时，通过率是1.0
276                                     # T=7时，通过率是0.99375
277 # t=8时，通过率是1.0
278                                     # T=8时，通过率是1.0

```

程序二：程序一对应相关系数热力图：

```

1  # -*- coding: utf-8 -*-
2
3  import pandas as pd
4  import seaborn as sns
5  import matplotlib.pyplot as plt
6  import matplotlib.colors as mcolors
7
8  path1 =r'D:\Mathematical_Modeling\training\rate.csv'

```

```
9
10 # 豪看
11 color1 = "#106391"
12 color2 = "#fbe5c3"
13
14 rgb_color1 = mcolors.hex2color(color1)
15 rgb_color2 = mcolors.hex2color(color2)
16
17 custom_cmap = mcolors.LinearSegmentedColormap.from_list('
    custom', [rgb_color1, rgb_color2], N=256)
18
19 df = pd.read_csv(path1)
20
21
22 plt.rcParams['font.size'] = 14
23 plt.rcParams['font.weight'] = 'bold'
24
25 plt.rcParams['font.sans-serif'] = ['SimHei']
26 plt.rcParams['axes.unicode_minus'] = False
27
28 plt.figure(figsize=(12,10))
29 sns.heatmap(df.corr(), annot=True, cmap=custom_cmap)
30 plt.xticks(rotation=18)
31 plt.title('相关系数热力图')
32 plt.show()
```