队伍编号：122

**2019西南石油大学数学建模竞赛**

**承 诺 书**

我们完全明白，在竞赛开始后参赛队员不能以任何方式（包括电话、电子邮件、网上咨询等）与队外的任何人研究、讨论与赛题有关的问题。

我们知道，抄袭别人的成果是违反竞赛章程和参赛规则的，如果引用别人的成果或其他公开的资料（包括网上查到的资料），必须按照规定的参考文献的表述方式在正文引用处和参考文献中明确列出。

我们郑重承诺，严格遵守参赛规则，以保证竞赛的公正、公平性。如有违反竞赛章程和参赛规则的行为，我们将受到严肃处理。

我们参赛选择的题号（从A/B中选择一项填写）： A

我们的报名参赛队编号： 122

成员姓名、院系及年级：1. 张宇轩 计科院2017级

2. 龚慧 理学院2017级

3. 杨强 计科院2016级

日期： 2019 年 5 月 4 日

**（此承诺书打印签名后作为纸质论文的封面，电子版与纸质版需保持一致。以上内容请仔细核对，如填写错误，论文可能被取消评奖资格。）**

红绿灯设置对城市道路通行能力的影响

摘要

红绿灯的设置会对道路的通行能力产生影响，特别是在十字路口，在下班高峰期时，某个相位的占用时间过短会降低整个路段的通行能力，进而使某个路段排起长长的队伍。在车辆稀少时，某个相位时间过长会导致资源的不合理分配，进而使人行道上出现很多等待行人。因此如何合理的利用已有资源，对每个路口红绿灯的持续时间进行合理分配，具有很大的实际意义与研究价值。

针对问题1，首先通过所给附件确定了当前情况下每个相位的红灯，黄灯，绿灯的持续时间。并通过相关交通法规，以及车辆相关标准确定了车头的最小间距，以及车流量的定义。假设在绿灯开始之前，当前相位路口的每个车道上存在着1000辆以最小间距来进行排列的等待通行车队。然后以红绿灯位置为原点，当前相位下车辆所排队的方向为x轴的正方向，建立**车辆位置动态模型**。该模型在考虑到每辆的加速过程，匀速过程，以及不同位置下车辆**启动延迟**时间等多种因素，以时间为自变量，对每辆车的动态位置进行**计算机仿真**，进而找到在一个绿灯周期后，位置过原点的车辆个数。最后由于车辆的加速度标准不同，我们给出了不同加速度下的动态车流量。得到在车辆加速度为2m/s2时的第一相位一个周期的绿灯可以通过30辆车，可满足的车流量为2232辆/小时；第三相位一个周期的绿灯可以通过11辆车，可满足的车流量为：864辆/小时。

针对问题2，首先引入**交通流**的概念，将道路中行驶的车辆看成流动的波，当第一相位路口由绿灯转向红灯时，则相当于在这个流动的波上突然阻塞，而阻塞的状态应该随着车流反向传播，使更多的车辆停止，进而引入**停止波**的概念。然后利用**格林希尔治-流-密度模型**，以及**格林伯-速度-密度模型**提出了两种**停止波速度**的计算方法，并比较在当前情况下的合理性，得到合理的停止波速度，再利用等待波速乘以等待时间得到等待队伍长度。接下来，在考虑**启动波**的情况下对模型1的车辆启动延时时间进行了优化。并利用改进后的模型1，给出了在第2车道发生事故时，事故后面的车的**变道算法。**利用计算机仿真，得到了不同事故发生位置下（可能是第2车道的车队中的第i，i+1辆车发生事故）的车流量。然后，根据流出量与到达量建立**道路阻塞模型**，用以评价当前道路的阻塞状况。最后以第4,5辆车发生追尾为例，评价其堵塞程度，并以车流量为自变量，阻塞指数为优化目标，建立阻塞指数优化模型，并有**单纯形法**进行求解，得到降低后的车流量为：753辆/小时。

针对问题3，首先为交通高峰，正常，以及车辆稀少等三种道路状况，定义车流量。引入**车辆平均等待时间**以及问题2中的道路阻塞指数，对不同道路情况下，红绿灯设置的合理性进行评价。并且在问题2的优化模型基础上，增加优化目标车辆平均等待时间，进而建立**双目标优化模型**。最后利用**遗传算法**进行求解，得到不同路况下的红绿灯设置。并对优化前与优化后的等待时间进行对比分析，发现不同路段减小的平均车辆等待时间为：（1）第一相位－45.58s（高峰）、－26.53s（正常）、0.62（稀少）；（2）第三相位118.93s（高峰）、39.24s（正常）、31.67s（稀少）。

**关键词：车辆位置动态模型 停止波 启动波 格林希尔治-流-密度模型 变道算法单纯形法 双目标优化模型 遗传算法**

# 一、问题重述

**1.1问题背景**

近年来，我国的国民经济水平不断提升，大量的人口涌进城市，城市的交通堵塞问题愈发严重，良好的控制红绿灯设置，是解决当前城市交通堵塞的重要措施之一。在城市交通网络中，各交叉路口是彼此相连的，通过交通信号灯实时的控制着车流信号，并对车流信号在网络中的运行状态进行分配。

交通便利对我们的日常生活至关重要。交通高峰时期，在很多车道会见到长长的车队等待红绿灯放行，红绿灯设置以及车道的指示可以使得拥挤的交通有序进行。有时候交通事故会占用车道，一条车道被占用，也可能降低路段所有车道的通行能力，即使时间短，也可能引起车辆排队，出现交通阻塞。

**1.2问题提出**

根据以上背景以及给出的两个附件，需要解决以下的三个问题：

1.求出附件一中第一相位一个周期的绿灯可以通过多少辆车；第三相位一个周期可以通过多少辆车；可以满足多大的车流量。

2.假设附件一中第一相位（自东向西车道）的车流量为1200辆/小时，试计算在第一相位变绿灯之前排队等候的车辆的长度。如果此时附件一中第➁车道出现了两车追尾使得此车道阻塞，试分析对该相位车流量的影响。车流量减少到多少才能使堵塞情况得到缓解。

3.假设附件一中东西方向车流量一直是南北方向车流量2倍，讨论在交通高峰、正常以及车辆稀少时现在的红绿灯设置是否能够满足要求，能否继续优化红绿灯设置使得车辆的等候时间尽量少。

其中，附件一给出了十字路口示意图，十字路口是一个双向六车道公路，给出了每个车道的行驶方向。其中东西向是主干路，限速70公里/小时，来去两个方向中间有隔离带，右拐的车道和直行的车道有很长的一段栅栏隔离，可以认为互不影响。南北向不是主干路，限速40公里/小时。因为车流量的不同，两个方向绿灯设置时间也不相同。附件二给出了各相位通行顺序，控制自东向西方向的路灯为第一相位，自西向东为第二相位，自南向北第三相位，自北向南第四相位。每相位前行、左转以及掉头的绿灯同时亮起，绿灯按照1,2,3,4相位的顺序亮起，周而复始。当某一相位的绿灯亮起时，其他相位都是红灯。

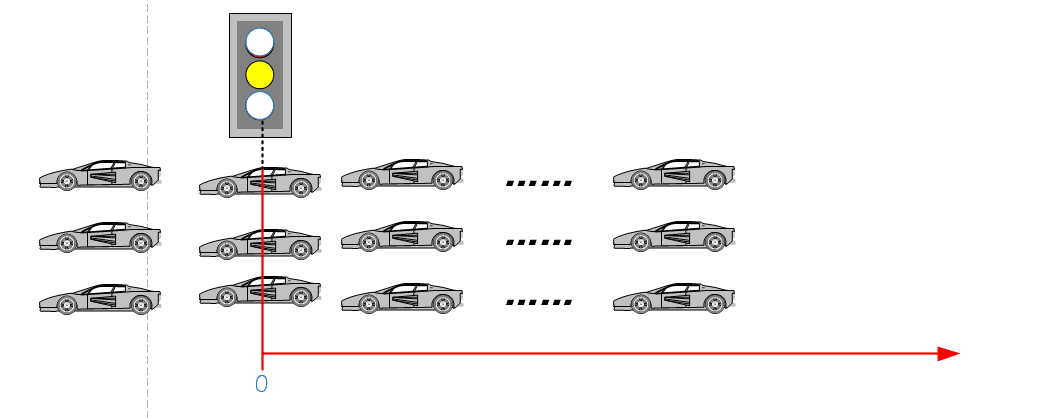
注：以上车辆只考虑四轮及以上机动车、电瓶车的交通流量，且换算成标准车当量数。

# 二、问题分析

**2.1问题一的分析**

问题一首先要求我们根据附件一中第一相位一个周期的绿灯通行时间来计算可以通过多少辆车，第一相位是控制自西向东方向的路灯，东西向是主干路，限速为70公里/小时，在一个周期内，能通行的时间为绿灯时间45秒，黄灯时间5秒，统计45秒内驶过红绿灯线以及第45秒时（即黄灯亮起时）车头在红绿灯线上的车辆数目，如图1所示，求出可以满足多大的车流量。第三相位是自南向北，南北向不是主干路，限速40公里/小时，在一个周期内，能通行的时间为绿灯时间20秒，黄灯时间5秒，统计20秒内驶过红绿灯线以及第20秒时（即黄灯亮起时）车首在红绿灯线上的车辆数目，如图1所示，求出可以满足多大的车流量。

**图1**

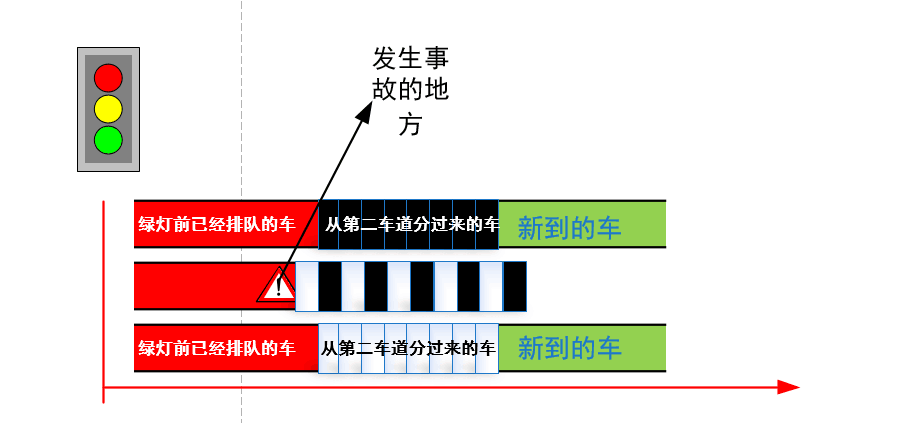


**2.2问题二的分析**

问题二首先提出要求改变了第一相位（自西向东车道）的主流量为1200辆/小时，由问题一求出的第一相位（自西向东车道）的主流量为2232辆/小时可知，车辆间距D增大，车流量减小，计算在第一相位变绿灯之前排队等候的车辆的长度，即计算5+45+5+（20+5）\*2=105秒时自西向东车道上排队等候的车辆的长度。

第二个问题是当第车道出现了两车追尾使得此车道阻塞，分析该条件对相位车流量的影响和当车流量减少到多少时才能使堵塞情况得到缓解。此时第、车道在原有的车辆基础上加上从第车道驶入的车辆以及新驶来的车辆，用N1、N2、N3分别表示、、车道上的车辆数目，L队表示原有的车辆长度，L表示车辆的长度，D表示车间距，是在绿灯时长内后方车辆会有多少辆新增车辆补充加入车队，D21、D23分别表示从第车道流入第、车道的车辆数目，则N1=L队/(L+D)+ + D21、N2=L队/(L+D) + - D21- D23、N3=L队/(L+D)+ + D23，如图2所示。根据已知条件可以求出车头时距：ht=3600/每条车道的车流量，进而求得一个周期内一条车道可以通过的最多车辆数目。

**图2**



**2.3问题三的分析**

问题三提出假设附件一中东西方向车流量一直是南北方向车流量的2倍，讨论在交通高峰、正常以及车辆稀少时现在的红绿灯设置是否能满足人们对于在一个周期内道路通畅的要求，然后继续优化红绿灯设置使得车辆的等候时间尽量少。可以运用遗传算法求阻塞率，然后计算两个堵塞因子X、Y，此时，可以看出在交通高峰、正常以及车辆稀少时现在的红绿灯设置能不能满足要求；进而得到适应度函数，通过适应度函数可以优化交通高峰、正常以及车辆稀少时的红绿灯设置，使得车辆的等候时间尽量少。

# 三、模型假设与符号说明

**3.1模型假设**

1. 车流速度为匀速，无外界影响
2. 车流过程畅通，没有意外事故
3. 忽略天气的影响
4. 信号灯正常工作
5. 忽略黄灯的影响，将其视为红灯
6. 假设汽车在刹车时存在一个平均刹车速度，既不考虑刹车加速度

**3.2符号说明**

**表1**模型数据变量表

|  |  |  |
| --- | --- | --- |
| **变量名称** | **代表含义** | **单位** |
| L | 汽车的长度 | m |
| D | 红灯时等待的相邻两辆车之间的距离 | m |
| T | 相邻两辆汽车启动的延迟时间 | s |
| a | 机动车的加速度 |  |
|  | 时刻t第n辆汽车所在的位置 | m |
|  | 表示第n辆汽车开始启动的时间 | s |
|  | 当前相位绿灯的时长 | s |
|  | 当前相位非绿灯时长 | s |
|  | m相位n辆车不考虑信号灯通过交通线时长 | s |
|  | m相位k车道n辆车通过交通线的所需的周期数 | s |
|  | m相位第n辆车考虑信号灯通过交通线的时长 | s |
| k | 车流密度 |  |
| Q | 单车道上交通量 |  |
|  | 区间内平均车速 |  |
| X | 有关堵塞指数的适应度函数因子 |  |
| Y | 有关绿灯时长的适应度函数因子 |  |
| Z | 适应度函数 |  |

注：未列出符号及重复的符号以出现处为准

# 四、问题一建模与求解

**4.1针对问题一的分析**

**对通行能力的分析**：

1. **理想的通行能力**

理想的通行能力市值在理想的道路与交通条件下的通行能力。

理想条件: <1>车道宽度合理

<2>侧向净宽充足

<3>车辆整体较大部分为市面上较为普遍的轿车类型

<4>驾驶员技术熟练，遵守交通法则

**2.情况假设根据实际生活中的一般情况假设当信号灯变为绿灯时道路的情况**

1. 在相位红灯周期中已经积累较长一段车辆于交通线前待信号灯变为绿灯时方可行使，且十字路口的车辆穿行秩序良好，不会发生阻塞；
2. 所有的车辆都是直行穿过路口，不拐弯行驶，并且只考虑马路一侧或单行线上的车辆；
3. 所有的车辆都相同，并且都是从静止状态匀加速启动；
4. 红灯时等待的每相邻两辆车之间的距离相等；
5. 前一辆车启动后，下一辆车启动的延迟时间相等。

**4.2问题一模型的建立**

根据题意可得出世纪的通行能力变化过程与车道的数目、车道宽度、行车的速度等变量有关系，因此建模过程中需要应用此类数据，根据题意需求得某一象限在一个周期内绿灯时能通过交通线的数量总和以及黄灯是车头已经通过交通线的数量总和。

**4.2.1汽**车**启动位置模型的建立**

假设红灯时车辆均停在交通线以内，以绿灯开始亮时为时刻t=0，由此求得汽车启动之前停车位置的模型为：

 (4.1)

用时刻表示第n辆汽车所在的位置，汽车的长度为L，红灯时等待的相邻两辆车之间的距离为D。

**4.2.2汽车启动时间模型的建立**

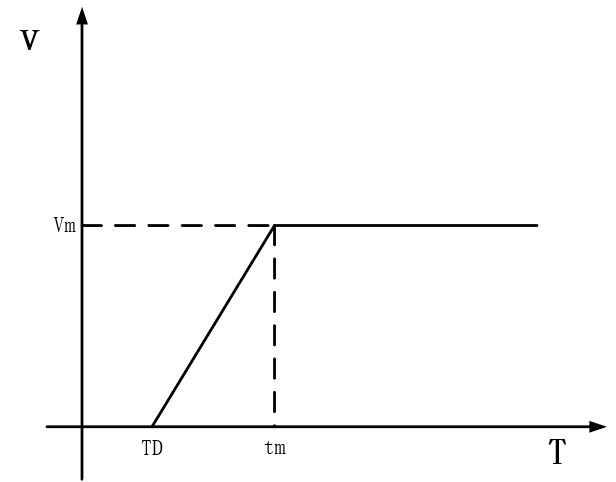
当信号灯由红定变为绿灯时车辆由静止开始启动。假设离交通线最近的司机以信号灯为基准，此后每位司机均以前一辆车的启动为基准开始启动。汽车启动的时间模型为：

 (4.2)

表示第n辆汽车开始启动的时间，T为司机的反应时间。

**4.2.3道路行驶模型的建立**

道路行驶模型应分为三段：



**图3**

**(1)车辆由静止开始启动**

车辆由静止开始启动但此时并未启动成功，因此此时的位置仍为初始位置：

 (4.3)

**(2)车辆启动完成后开始做匀加速直线运动直至速度达到当前车道允许的最大时速**

汽车刚起动时应该按照匀加速的规律运动，汽车启动后在时刻t (t>)，a表示机动车的加速度，车辆所处的位置应为：

(4.4)

**(3)车辆达到最大时速后，以最大时速保持匀速直线运动**

车辆到达最大速度后以保持运动直至绿灯时长结束，汽车启动后在时刻t (t>)，车辆所处的位置应为:

 (4.5)

tM表示达到道路最大限速所需要的时间，VM是道路最大限速。

**4.2.4车辆位移坐标体模型建立**

综合上面的分析，我们就得到了汽车在道路上行驶的模型为(4.6)

**4.2.5每条车道上通行量模型：**

 (4.7)

用来统计第k条车道上第n辆车是否能通过交通线, k=1,2,3

n=1,2,……1000.

**4.2.6每条车道上车流量模型：**

 (4.8)

 (4.9)

为第k条车辆的总车流量，其中k=1,2,3

n=1,2,…….（为第k条车道的车辆数）

**4.2.7总模型的建立**



**4.3问题一模型的求解**

由于每个车道的情况相似，所以只需要求出一条车道的车辆最大容量之后，再将这个最大容量乘以3得到总容量。首先我们需要初始化N\*M的矩阵，N表示总时间45秒，M表示当前车路状况（排队车辆数），为了能够求得最大通过量，我们将整个队列设置为1000，在一个周期后，找出这1000量车辆中所在位置小于0的车辆个数，则为通过红绿灯口的车辆个数。

主要步骤如下:

Step1：初始化车辆长度，车辆间距车辆加速度等具体的值。

Step2：初始化每辆车初始的位置的水平坐标，以及反应延迟时间

Step3：根据问题1所建立的模型进行递推求解，得到每一时刻每一辆车的具体位置

Step4：统计第45秒时，车辆水平坐标值小于0的车辆数之和，进而得到3车道下通过的车辆总和。

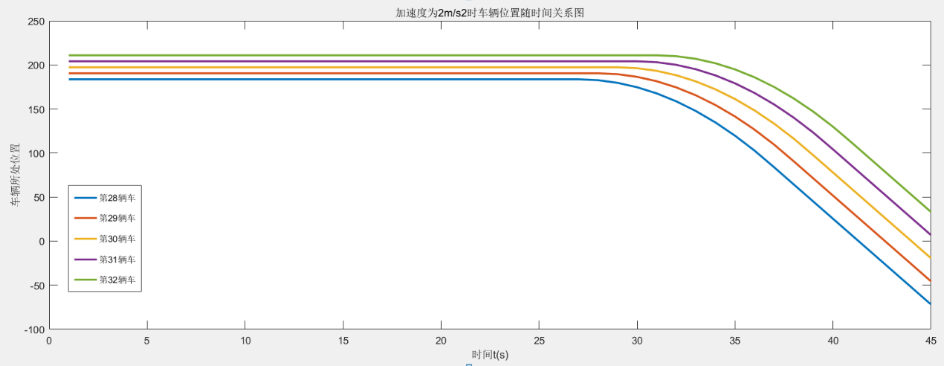


**图4** 流程图

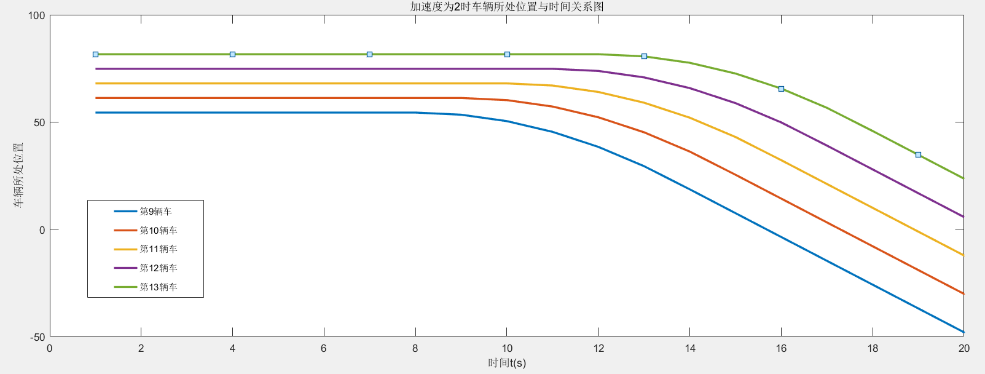
**4.4结果分析**

关于模型的参数值，我们取L=4.8米，D=2米，T=1秒。在东西方向即1、2相位上最大速度为70公里/小时。在东西方向即3、4相位上最大速度为40公里/小时。加速度我们将取由到以每次0.1的间隔递增对每个相位中能通过的最大车的数量惊进行分析，进而求得车流量。

**图5** 第一相位，当a=2m/s2时抽取部分车辆的位置坐标

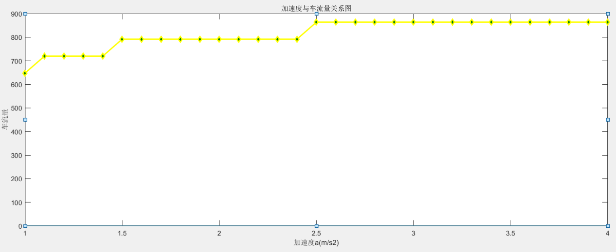
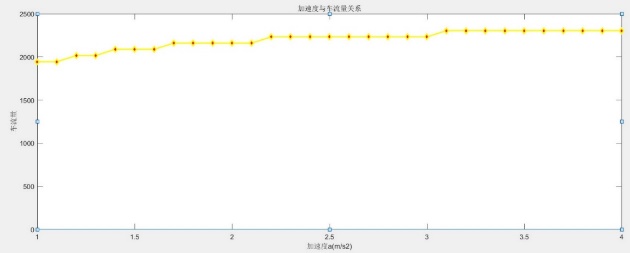


**图6** 第三相位，当a=2m/s2时抽取部分车辆的位置坐标



根据图5和图6可知，当a=2m/s2时绿灯时间45秒内道路的最大通行车辆为30辆；当a=2m/s2时绿灯时间20秒内道路的最大通行车辆为11辆。可以清晰的看到在绿灯时间内，当车辆数目逐渐增加时，车辆离红绿灯线的距离逐渐减小，最后有一些车辆在周期内越过红绿灯线，一些卡在红绿灯线上，还有一些还没来得及到达红绿灯线绿灯时间就停止了。

如图所示为，不同a下，第一、三相位一个周期的车流量：

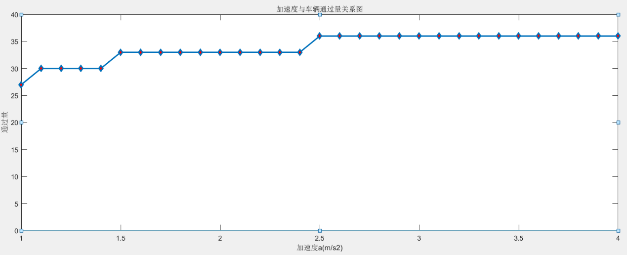
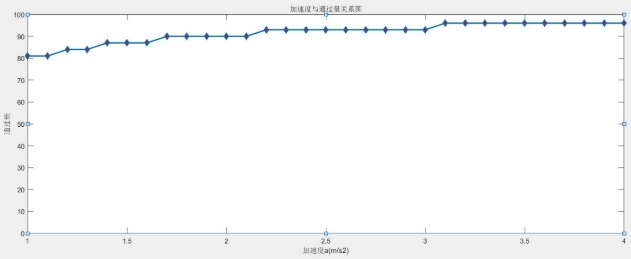


**图7**第一相位一个周期的车流量

**图8**第三相位一个周期的车流量

由图7和图8可知，不同a下，第一相位一个周期的车流量随着a的增大不断增大，最后可以满足的最大车流量为2232辆/小时；同样，第三相位一个周期的车流量也随着a的增大不断增大，最后可以满足的最大车流量为864辆/小时。直观的结果显示，在车道内，车辆从初速度0加速到最大限流速度后，速度不再变化，车流量也达到稳定值，在加速过程中，随着a的增大，速度增大，车间距增大，可以容纳更多的车辆通过，从而快速增加车流量。

如图所示为，不同a下，第一、三相位一个周期的绿灯可以通过车辆数：



**图9**第一相位可以通过车辆数

**图10**第三相位可以通过车辆数

根据图9和图10可观察到的信息是，不同a下，第一相位一个周期的绿灯（45秒内）可以通过车辆数为96辆；第三相位一个周期的绿灯（20秒内）可以通过车辆数为36辆。在一个周期内，加速度逐渐增大，当达到车辆在该道路的最大速度时，速度不再变化，此刻，车流量达到最大值，可通过车辆数目也达到最大值，不再变化，最后统计一个周期内该道路可通过的所有车辆数。

# 五、问题二建模模型建立与求解

**5.1基于交通波理论的问题分析**

交通波理论描述同向运动的两股不同状态交通流相遇时状态转移的过程。该理论最初从流体力学发展而来，常常被应用于排队形成与与消散的研究，在上述问题中我们仍需对道路情况进行理想化模拟，如与模型一的理想化情况模拟一致。

理想的通行能力市值在理想的道路与交通条件下的通行能力。

理想条件: <1>车道宽度合理

<2>侧向净宽充足

<3>车辆整体较大部分为市面上较为普遍的轿车类型

<4>驾驶员技术熟练，遵守交通法则

此外此模型需要考虑停车波的问题，并以此来建立模型。但前面一辆车刹车导致后一辆车跟随刹车，但由于司机存在反应的时间间隔由此产生停车波，启动波同理。

**5.2模型一：停车波与启动模型**

**5.2.1传统的停车波与启动模型**

**格林希尔模型：**

Step1：建立停车波波速公式：

 （5.1）

为车辆阻塞时的密度，为车辆正常行驶时的自由流密度

Step2：令，则称为标准化密，则有:

和 （5.2）

为绿灯起始时的速度，为绿灯结束时即黄灯开始时的速度

Step3：将以上两速度带入波速公式求得:

 （5.3）

、分别为绿灯起始和结束时车辆的密度。将上式整理可得

 (5.4)

Step4：假定车流的标准化密度，在区间内以作为平均速度行驶。现模拟行驶到交叉停车线处遇到黄灯或红灯车辆停止，此时=1，根据(1)推导出以下的停车波模型

 (5.5)

 (5.6)

由于信号灯的变化导致车辆停止行驶，且因为人的反应需要时间所以产生停车波，且停车波将以的速度向后方传播，经过t秒后，将形成一列长度为的排队车队

**5.2.2改进的停车波和启动波模型**

**格林伯速度-密度模型**

Step1：根据题意建立格林伯速度-密度模型核心公式:

 (5.7)

上式中: 为汽车在行驶时的最佳速度，即交通流能达到通行能力时的速度，保障交通不拥堵。

Step2：对于停车波，令，带入格林伯速度-密度模型结合波速公式有:

 (5.8)

由此求得，情况下的结合波速度，在上式中为停车密度

式(3)即为采用格林伯速度-密度模型修正的停车波模型

Step3：利用格林伯速度-密度模型对启动波模型进行修正，修正方法同前。

Step4：此时令，，修正后的模型形式如下:

 (5.9)

 (5.10)

上式中为启动密度，即车队启动时波阵前面的密度

Step5：建立停车时车辆排队长度模型

 (5.11)

、分别是停车波开始产生和结束时车辆的速度、分别是停车波开始产生和结束时的时间

**5.2.3两个模型的优劣分析:**

1. 两个模型均以停车波与启动波作为模型较好的模拟了现实中公路的情况
2. 格林伯模型解决了格林希尔模型因后方堵塞密度较大而产生的与实际问题不符的情况
3. 只能单一的表示车道的速度-流量密度关系，并非高速公路上观测的数据
4. 当堵塞密度较小时格林伯模型对实际问题的契合度较小，此时效果不如格林希尔模型

**5.2.4总模型的选择**

现分析车队的启动过程，当堵塞车队长度较长时，信号灯由红灯变为绿灯，车队开始行驶且有头部开始产生启动波，虽然启动波波速>停车波波速，但启动波并不能立即与停车波产生的效果抵消，后方停车波仍在产生且阻塞密度仍然很大。但是当堵塞车队长度基数较小时，可当做启动波效果与停车波立即抵消。所以改进后的格林伯模型与实际问题的契合度较小所以在此条件下应采用格林希尔模型而不能采用格林伯速度-密度模型。

 (5.12)

此外车头间距、车头时距及速度三者关系

 (5.13)

其中为前后车间的车头间距，为前后车间的车头时距，为汽车行驶车速：

 (5.14)

上式中为车头时距，Q为单个路段车流量

 (5.15)

为车流密度，区间内的平均车速

综上，基于格林伯模型推导出的停车波和启动波模型比传统的停车波和启动波模型更接近实际交通流运行状况，因此我们选择格林伯速度-密度模型进行建模对问题进行求解。

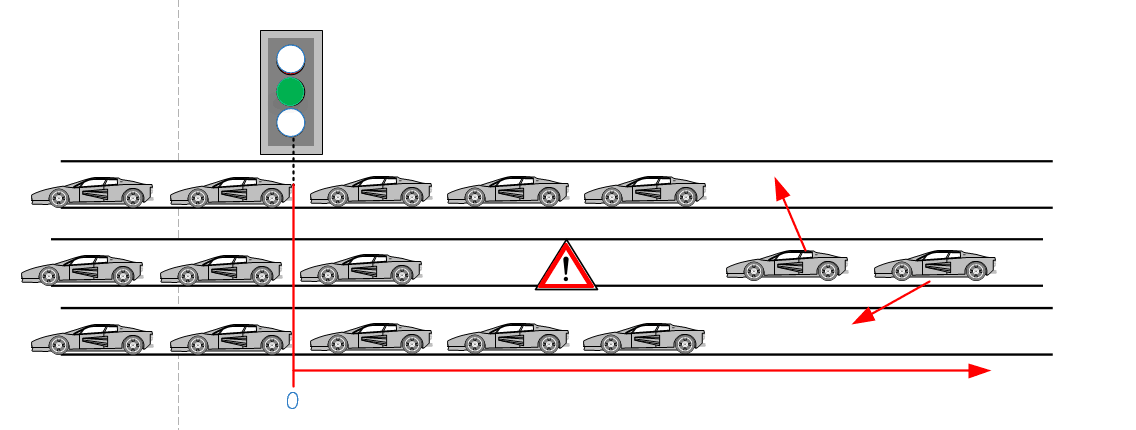
**5.3模型二：基于堵塞后的堵塞指数模型**

**5.3.1模型建立与求解**

Step1：当绿灯亮起，车辆开始行驶，启动波波速，此时从车头产生启动波，以格林伯速度-密度模型做为基础对题干第一问中的司机反应时间T进行优化可得

 (5.16)

Step2：当发生追尾事故后2号车道事故发生地点后的车辆驶入1、3车道因此三条车道的交通量均发生改变。路段交通情况具体如下：



**图11**

Step3：事故发生后，事故发生路段车辆向两侧行驶，且会有新的车辆加入进来，事故发生后交通路段模型与图示如下

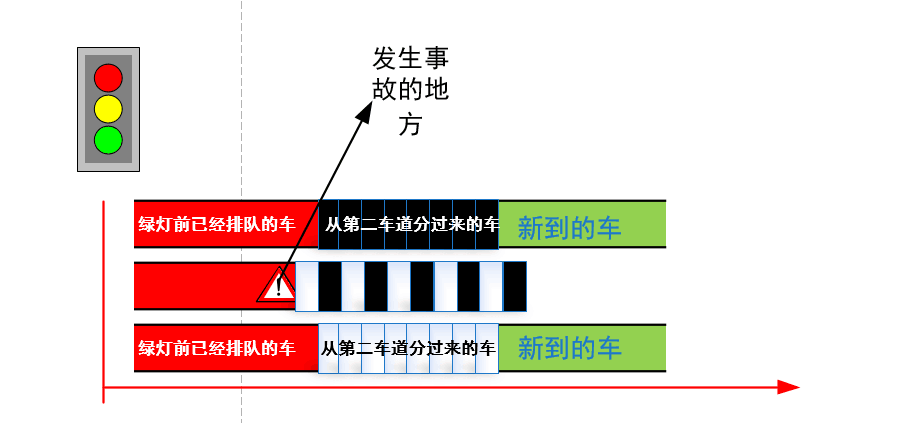


(5.17)

(5.18)

(5.19)

 (5.20)



**图12**

是由2号车道驶入1号车道的车辆，是由2号车道驶入3号车道的车辆，是在绿灯时长内后方车辆会有多少辆新增车辆补充加入车队，应为绿灯时长/车头时距。

在2号车道发生追尾事故后此时的车流量模型为：

， (5.21)

大体模型与第一题模型一一致，只是对时间间隔进行优化其中k=1、2、3

N=1、2……

所以在撞车后新的车流量模型如下

  (5.22)

**5.3.2撞车后车流量的影响模型**

综上所述我们由模型一的建立结合格林希尔模型建立了发生交通事故后该路段的道路拥挤状况模型。

其中为第k条路段当前的所有积累的车辆总数，为当前路段内的队伍长度，为之前格林希尔模型中车辆正常行驶时的自由流密度，为标准化密度，为时间。

由此算出新的路段内各个道路的车流量以及总的车流量

**5.4模型三：道路阻塞情况优化总模型**

**5.4.1问题分析:**

交通灯的作用在于能通过红灯及绿灯的时间来疏通十字路口处方向车道的车流。要保证在交叉路口出没有车流堵塞，各方向的交通灯时间就必须使得车道上车流通过且不影响其他方向上的车流。因此，要从两个法方面考虑该问题.第一方面:首先考虑当绿灯开始到结束时交通线能通过的车辆称为到达量，以及车辆在上一个红灯周期内积攒的车辆称为已有辆。第二方面:考虑从信号灯变为绿灯开始到结束时通过交通线的车辆称为流出量。则利用以上变量可以逐渐建立起一个有关堵塞指数的函数关系模型。

**5.4.2问题假设**

1. 不考虑交通灯黄灯的时间
2. 假设汽车在刹车时存在一个平均刹车速度，既不考虑刹车加速度

**5.4.3模型的建立与求解**

  (5.23)

(5.24)

 (5.25)

代表上一个周期以前交通线内红灯积攒的车辆数，，表示一个周期内非绿灯的时长，为之前问题一的车辆位置模型通过交通线的车辆数总和

表示当前周期绿灯时间段内通过交通线的车辆数目，为当前道路的出流量。

综上所述，需求得车流量与的关系。

由以上模型可算得堵塞指数：

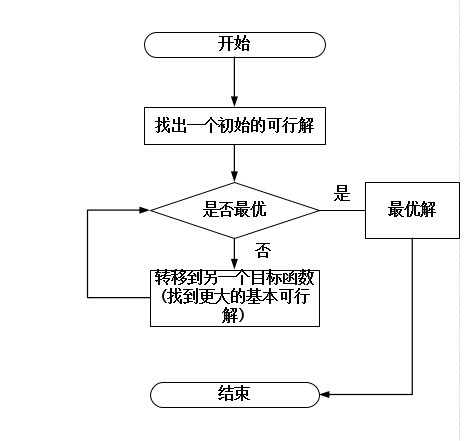
当>1时则此时道路堵塞，且越大说明堵塞程度越大；

当<1时则此时道路不堵塞，且越小说明堵塞程度越小。

**5.4.4道路阻塞情况优化总模型**

以上建立模型中运用到运筹学中的单纯行法，即一般线性规划问题中当线性方程组的变量数大于方程个数，这时会有不定数量的解，而单纯形法是求解线性规划问题的通用方法。 具体步骤是，从线性方程组找出一个个的单纯形，每一个单纯形可以求得一组解，然后再判断该解使目标函数值是增大还是变小了，决定下一步选择的单纯形。通过优化迭代，直到目标函数实现最大或最小值。 换而言之，单纯形法就是秉承“保证每一次迭代比前一次更优”的基本思想：先找出一个基本可行解，对它进行鉴别，看是否是最优解；若不是，则按照一定法则转换到另一改进后更优的基本可行解，再鉴别；若仍不是，则再转换，按此重复进行。因基本可行解的个数有限，故经有限次转换必能得出问题的最优解。如果问题无最优解，也可用此法判别。

**图13** 单纯行法算法图



**目标函数：**

 (5.26)

***约束条件：***



由模型分析可知是的函数本质由决定。当道路发生事故后较大使得道路表现为堵塞。所以现在将减小的值使得减小，寻找使得1时的的值即为问题所求的车流量应减小到何值使得道路拥堵状况得到缓解,在此模型中目标函数为

 （5.26）

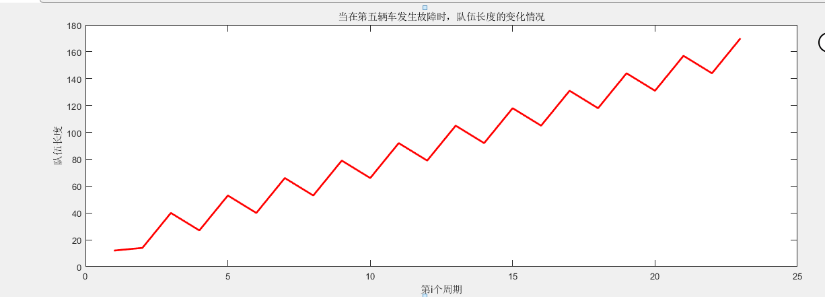
所以需要找到 Z的左极限接近0时的值，则此时，的值为为从堵塞状态到刚好不堵塞状态的临界值，即为我们所求的缓解拥堵后的车流量。

其中，D为车辆间隔，L为车身长度C:\Users\gh\Documents\Tencent Files\2270368329\Image\C2C\(%4I})3D90S(YCHYUMM_~H5.png是在绿灯时长内后方车辆会有多少辆新增车辆补充加入车队，应为绿灯时长/车头时距，a表示机动车的加速度，以绿灯开始亮时为时刻t=0，时刻表示第n辆汽车所在的位置，表示第n辆汽车开始启动的时间,tM表示达到道路最大限速所需要的时间，VM是道路最大限速，为第k条车辆的总车流量，其中k=1,2,3，n=1,2,……1000，用来统计第k条车道上第n辆车是否能通过交通线, k=1,2,3。

**5.5结果分析**

我们模拟了当第5辆车发生故障时道路状况

**图14** 堵塞前道路车辆状况图

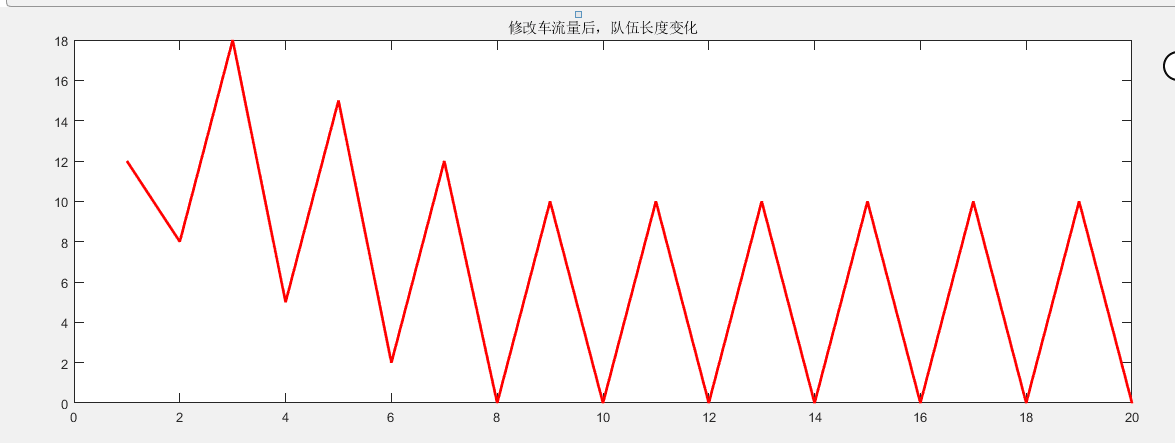


由图14分析可知道路拥堵呈现一个周期变化由于到达量大于流出量所以队伍在多个周期内呈现越来越长的趋势，故造成道路越来越拥挤。

利用模型进行求解:

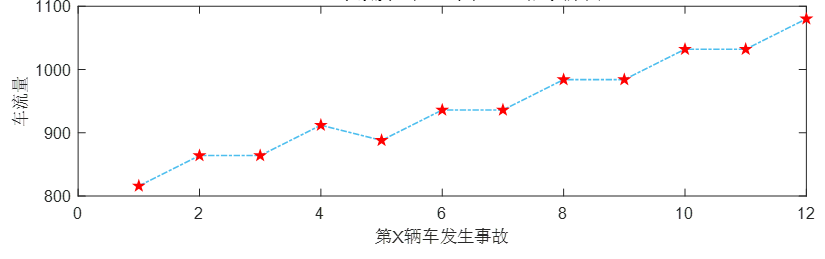
由上图14可知当第5辆车发生故障时 现阶段道路阻塞系数较大，呈现越来越拥挤的现象。现对目前的车流量进行计算机仿真处理求得当车流量Q为753(辆/h)时为上述单纯形法函数模型的最优解

**图15** 此时模拟出的道路情况图



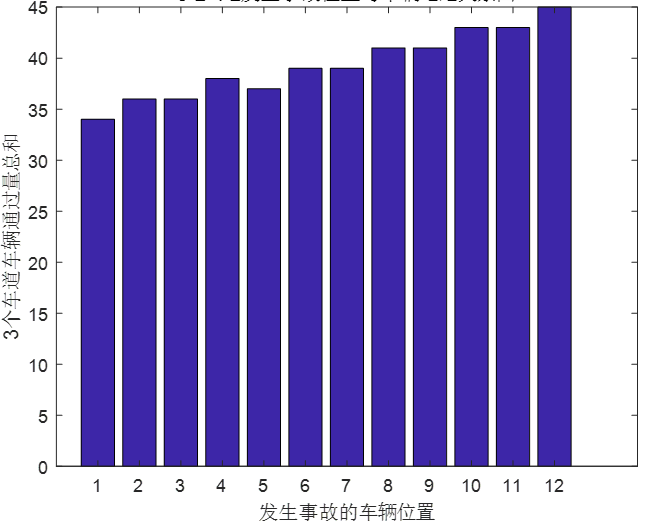
由图15可知在短暂的几个周期后车辆流出量慢慢的抵消堵塞时的积累量与新增车辆到达量的总和，由此使得道路拥堵状况得到缓解。

**图16** 事故发生位置与车流量对应关系图



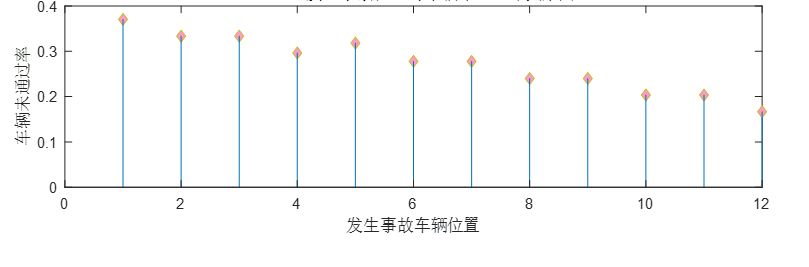
由图16可知，在车辆发生事故的条件下，一个周期内越往后的车辆发生事故，则此时瞬时车流量越大，较事故位置发生靠前的情况整体通行的模型破坏的越少，越有利于通行，则如果发生事故位置靠前则对整体的通行模型破坏较大越不利于通行。即随着X（车辆即将通过红绿灯线的个数）的增大，车流量越大。

**图17** 发生事故位置与车辆通过关系图



在图17中可以看到，当第12辆车发生事故时，一个周期内该道路车辆通过量总和数最大为45；当第1辆车发生事故时，一个周期内该道路车辆通过量总和数最小为34。当发生事故的车辆在一个周期内的位置越往后，对该时间通过车辆个数的影响越小，即发生事故车辆位置越靠后，一个周期内该道路车辆通过量总和越大。

**图18** 发生事故位置与车辆未通过率关系图



由图18可观测到的信息是，第12辆车发生事故时，一个周期内该道路车辆未通过率最小为0.1667；第1辆车发生事故时，一个周期内该道路车辆未通过率最大为0.3703。当车辆发生事故的时间处于一个周期内绿灯时间快结束时，对该时间车辆通过率的影响越小，即发生事故车辆位置越靠后，一个周期内该道路车辆通过量总和越大，车辆未通过率越小。

# 六、问题三建模与求解

**6.1问题分析**

交通信号灯的设计问题，主要是缓解交通堵塞，从而达到时间优化、安全性最高的目的

遗传算法是在研究自然遗传现象与人工系统的自适应行为时，模拟生物进化现象，并采用自然进化机制来表现复杂现象的一种全局群体搜索算法。

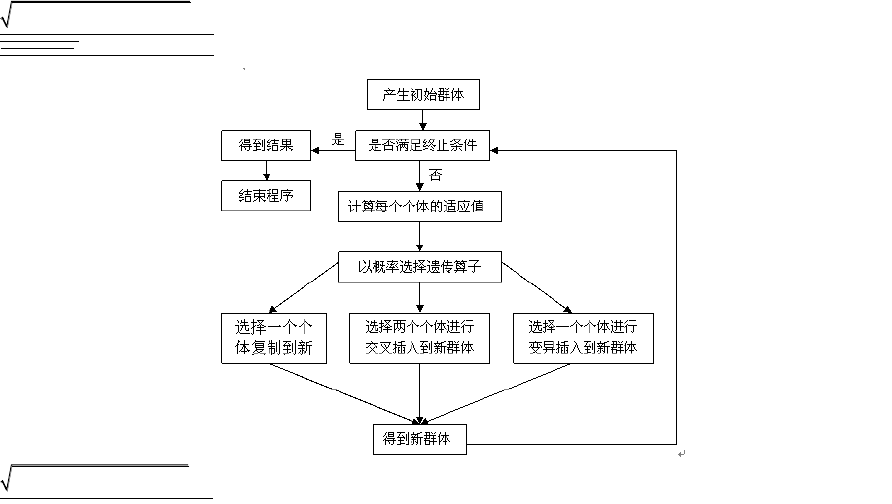
**6.2模型的建立与求解**

本题用到遗传算法来进行求解首先确定出关键参数:基因编码、适值函数。

**遗传算法简介**

遗传算法是计算数学中用于解决最佳化的[搜索算法](https://baike.so.com/doc/6058609-6271658.html" \t "_blank)，是进化算法的一种。进化[算法](https://baike.so.com/doc/2758411-2911336.html)最初是借鉴了进化生物学中的一些现象而发展起来的，这些现象包括遗传、突变、自然选择以及杂交等。遗传算法通常实现方式为一种[计算机](https://baike.so.com/doc/3435270-3615253.html)模拟。对于一个最优化问题，一定数量的候选解（称为个体）的抽象表示（称为染色体）的种群向更好的解进化。

**图19** 遗传算法大体流程图



**6.2.1种群初始化**

在红绿灯的时间优化模拟问题上将四个相位的绿灯时间(m=1,2,3,4) 和(m=1,2,3,4)表示黄灯时长作为算法中的基因编码首先根据红绿灯时长周期来初始四个相位的绿灯时长，并根据各个相位的绿灯时长以及之前所建立的停车等待车队长度模型计算出当前该相位车道内的停车数量与长度。

由于整体模型中相位一、二红绿灯周期时长相同，相位三、四红绿灯周期时长相同，故在此只分析相位一、三的情况。

**6.2.2适应度函数**

适应度表明个体或解的优劣性，不同的问题，适应度函数的定义方式也不同，本题目中适应度函数根据堵塞指数和每条道路上的车辆平均等待时长，并根据题目要求确立二者的函数关系式作为适值函数来对之后由不同绿灯时长产生的结果进行选择

对红绿灯的时长是否合理进行评判首先要运用到之前的堵塞指数模型来作为标准

Step1：现令来评判红绿灯时长情况，且越小越好，为有关堵塞指数的适应度函数因子。

Step2：模拟不同道路上车辆的平均等待时长作为另一个适应度函数的评判标准，为m相位第n辆车考虑信号灯通过交通线的时长。

m为不同的相位(m=1,2,3,4)， n为当前道路上的第n辆车(n=1,2……)。

则要求需要用到之前建立的车辆位置模型

(6.1)

来计算当前车道上第n辆车通过红绿灯需要的时间

时刻表示第n辆汽车所在的位置，表示第n辆汽车开始启动的时间。

Step3：令求得当前m相位上第n辆车不考虑红绿灯通过交通线的时间即为为等待时间。

Step4：由  (6.2)

和  (6.3)

算得当前m相位上第n辆车考虑红绿灯时通过交通线的时间即为等待时间。其中为当前相位绿灯的时长为当前相位非绿灯的时长。表示当前相位第k条车道的第n辆车通过交通线需要的周期数。

为m相位k车道n辆车通过交通线的所需的周期数，为m相位第n辆车考虑信号灯通过交通线的时长，t(m,n)为m相位n辆车不考虑信号灯通过交通线时长，为当前相位非绿灯时长。

Step5：令  (6.4)

作为适应度函数的另一个衡量标准

双目标优化模型目标函数：，

令进行求解

**6.2.3基于遗传算法红绿灯时长优化模型**

 (6.5)



Step6：令作为最终的适应度函数来对不同的绿灯时长进行拟合使得远大的值则越符合期待，并最终找到最优解。

为有关堵塞指数的适应度函数因子，时刻表示第n辆汽车所在的位置，表示第n辆汽车开始启动的时间，为m相位k车道n辆车通过交通线的所需的周期数，为m相位第n辆车考虑信号灯通过交通线的时长，t(m,n)为m相位n辆车不考虑信号灯通过交通线时长，为当前相位非绿灯时长。

**6.2.4选择操作**

选择操作的目的是为了从当前群体中以一定的概率选择优良个体到新群体中，将选择算子作用于群体，从而使优化的个体有机会直接遗传到下一代或通过配对交叉产生新的个体再遗传到下一代，被选中的概率与适应度值有关，适应度值越大，被选中的概率也越大。在本题中将对绿灯时长进行不同的拟合进而求出相对应的个体的适应度值。之后再进行选择操作。

**6.2.5 交叉操作**

交叉操作是遗传算法中最主要的遗传操作，通过交叉操作可以得到新一代个体，新个体结合了父辈个体的特性，交叉体现了信息交换的思想，利用不同映射杂交，确定交叉操作的父代，将父代样本两两分组哦，每组重复以下过程

以表示第n次的绿灯时长模拟量的染色体，以表示第n+1时的绿灯时长模拟量的染色体

则交叉互换将由以下染色体



变为



由此产生交叉互换，交叉是希望不同的个体在产生下一代是，能够结合自身的优势基因，产生更好质量的下一代。

**6.2.6 变异操作**

变异操作可以堪称是外界对种群的影响。变异是为了引入新的因素希望个体在外界的作用下，能够实现自我优化，生成好的基因。

例如以下基因

可能变异为

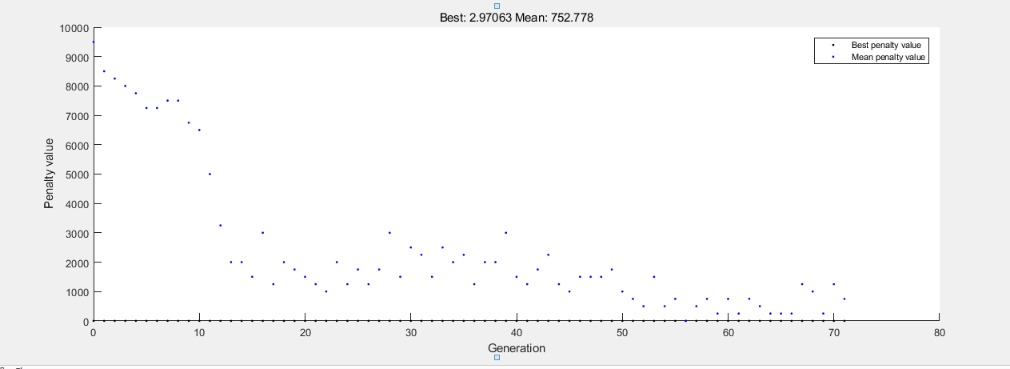
这种变异突破了原有的红绿灯周期，和题目中的相对相位的红绿灯周期时长相等的情况，可能更优，可能更劣。

**6.3结果分析**

根据遗传算法的大体流程，我们拟合了高峰期、正常期、稀少期三个时期根据信号灯时长变化产生的车辆平均等待时长。其中高峰期车流量为3600辆/h,正常期车流量为2000/h,稀少期车流量为1000辆/h。

其中遗传算法最优解的趋势图如下：

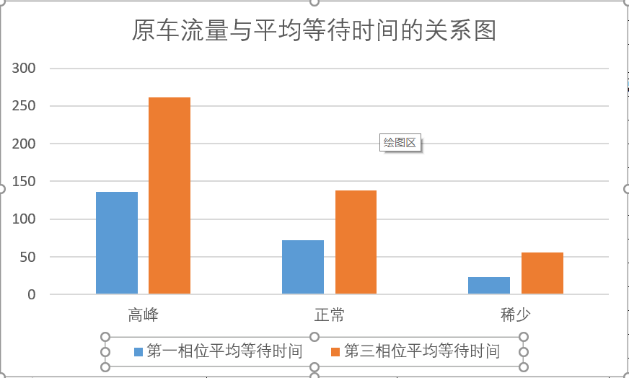
**图20** 遗传算法最优解的趋势图



**当信号灯时长优化前：**

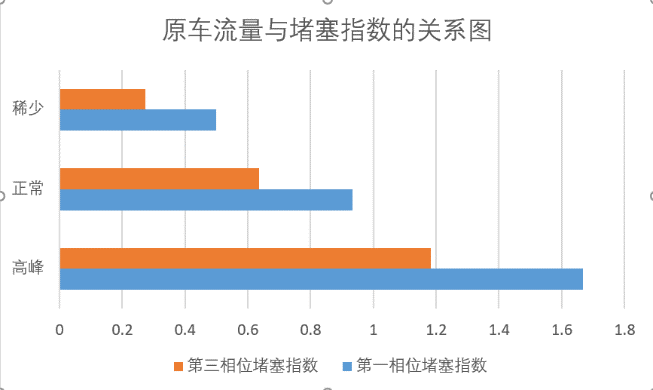
此时的一、三相位内的车辆平均等待时长与车流量产生的关系图如下：

**图21** 原车流量与平均等待时间的关系图



则此时车流量与堵塞指数的关系图为：

**图22** 车流量与堵塞指数的关系图

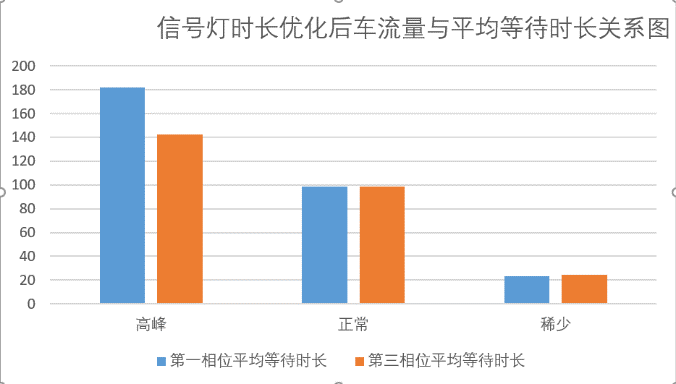


由图22可见，在原信号灯红绿灯时长分配情况下稀少、正常时期堵塞指数均小于1均存在时间富裕，这也就产生了时间浪费。而在车辆高峰时堵塞指数均大于1，使得车辆通过交通线的平均等待时间较长这又存在着较大的时间不足。由此可知原信号灯时长分配的合理性较差。

**当信号灯时长优化后:**

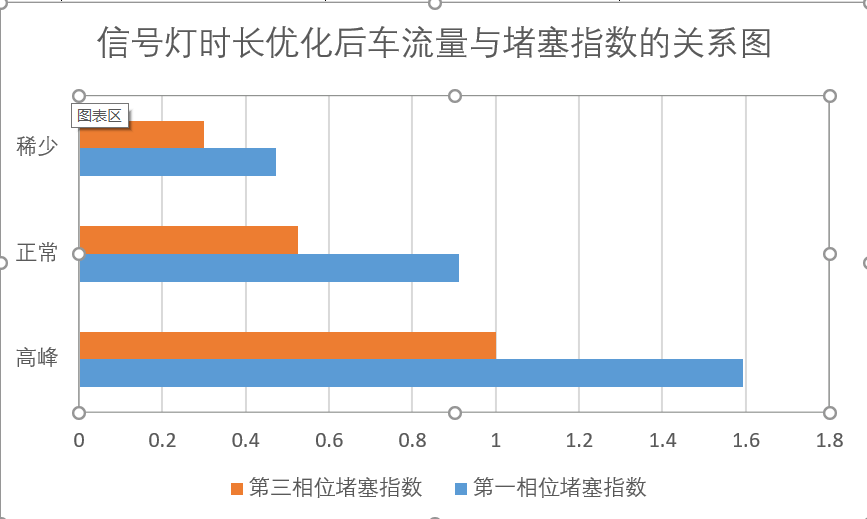
控制高峰、正常、稀少期的车流量不变与优化前的信号灯时长形成对比。则此时的一、三相位车流量与平均等待时长的关系图为

**图23** 信号灯时长优化后车流量与平均等待时长的关系图



与优化前的车流量与等待时长关系图（图21）进行对比，发现一、三相位在三个时期的平均等待时长均有所降低，且原第三相位平均等待大于第一相位平均等待时长，呈现出第三、四相位更为拥堵的情况，而优化后的一、三相位等待时长基本持平，很好的改善了以上问题。

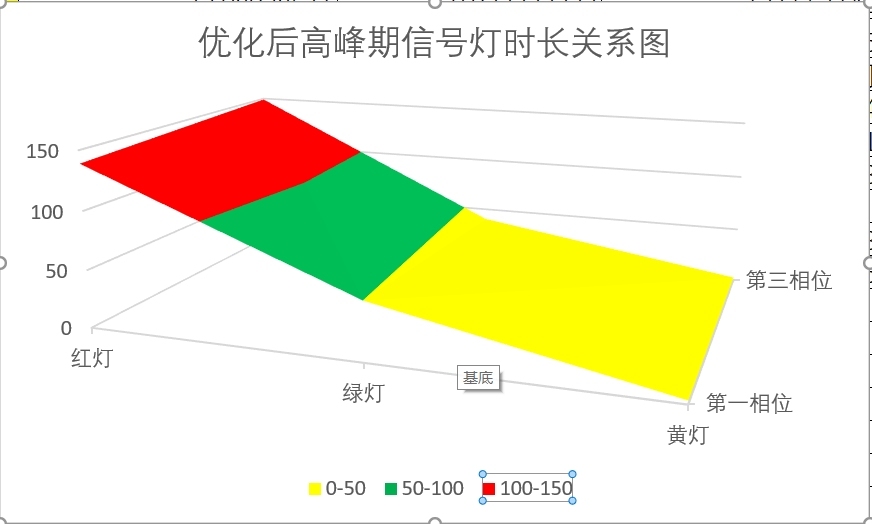
**图24** 信号灯时长优化后车流量与堵塞指数关系图



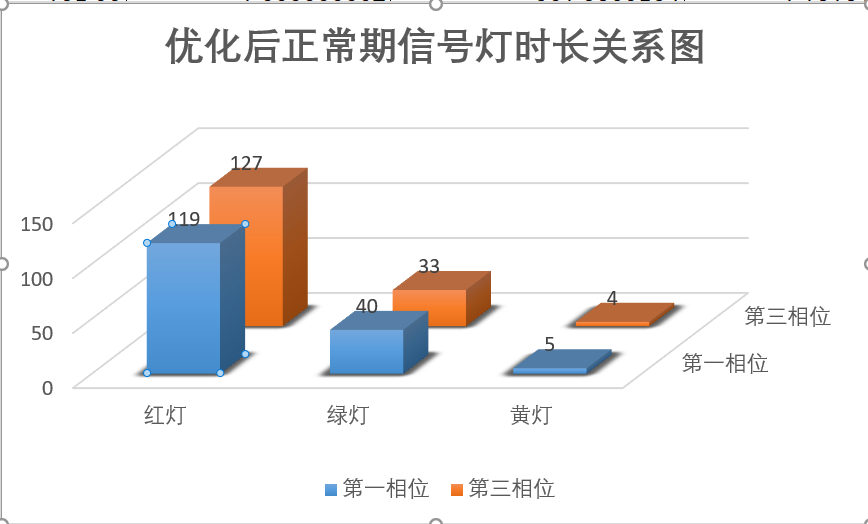
由图24可知，三个时期的的堵塞指数较优化前均得到了降低，缓解了道路堵塞情况。且除了高峰期的第一相位仍存在一定量的堵塞情况，其他时期的其他道路的堵塞指数均小于1且等待时间均得到了改善。由此可知此基于遗传算法的信号灯时长优化模型对于改善不同时期的道路拥堵状况有着较好的表现结果。

以下是优化后的根据车流量的不同所产生的信号灯时长不同的关系图：

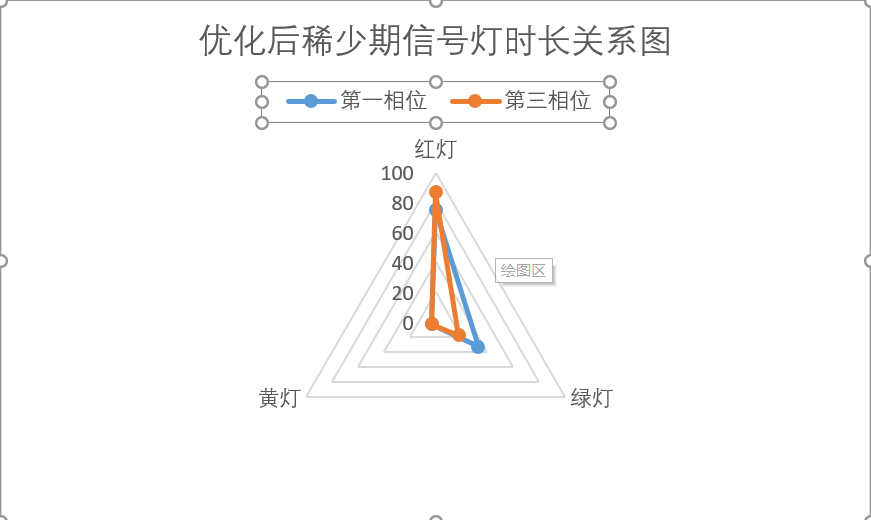
**图25** 优化后高峰期信号灯时长关系图



**图26** 优化后正常期信号灯时长关系图



**图27** 优化后稀少期信号灯时长关系图



由上图分析可知当车辆的堵塞指数较小时应当减小信号灯时长总周期以及相对应的绿灯时长来对道路车辆的时长浪费情况进行改善。

当车辆的堵塞指数较大时应当扩大信号灯时长总周期以及相对应的绿灯时长来对道路车辆的堵塞情况进行改善。

# 七、模型评价与优缺点

**7.1问题一的模型**

**模型的评价**

**优点：**

1. 利用我们这个方案，可以很合理的定位出当前信号灯内等待的车辆数，并具有极高的可靠性。
2. 这个模型方案具有极高的操作性，比较容易理解
3. 模型的可操作性强，适用范围广泛，得到的因素权重可信度比较高。
4. 考虑到多种加速度产生的结果，针对不同加速度算得不同的车辆流，结果不单一有较强的说服性，试用性较广。

**缺点：**

1. 模型较为理想化，而真正的红绿灯规则还要考虑具体情况。
2. 对于车长、间距、反应时间较为统一，实际情况则无法如此的统一化
3. 主要以常用轿车作为主体进行建模，没有考虑小型车和大型车的影响。

**模型的改进:**

应考虑道路实际情况存在不同类型的车辆从而产生的车辆相对间距不同的情况，通过市场调查以及大数据分析得到相对应的新的车辆相对间距模型。从而对模型一进行改进，达到更加优化的目的。

**7.2问题二的模型**

**模型的评价**

**优点：**

1. 建立的模型能与实际紧密联系，停车波，启动波、堵塞指数模型都符合实际情况，使得模型的可信度较高，所求的数值准确度高，通用性、推广性较强。
2. 基于堵塞指数模型考虑到的情况相对全面，仿真结果合理性较强。
3. 根据计算机仿真模拟出不同位置发生事故造成堵塞的情况，算得数据较全。

**缺点：**

1. 基于(道路堵塞指数)模型中的参数确定的随机因素较多，适当的模型不能将其准确地反映出来。
2. 模型较为理想化，而真正的红绿灯规则还要考虑具体情况。

**模型的改进:**

在求解如何缓解堵塞情况时建立的堵塞指数模型没有考虑到因为撞车而产生的一定程度小型的停车波的产生。这一定程度上会影响结果的准确性。若要提高模型的精度，应该把因撞车产生的停车波的变化考虑进去。

**7.3问题三的模型**

**模型的评价**

**优点：**

1. 解决现实生活中的部分红绿灯时间问题时，利用我们这个方案，可以很方便地定出合理了红绿灯时间，并具有极高的可靠性。
2. 该方案可以减少实际情况中的拥堵问题，加速车辆流通。
3. 这个模型方案具有较高的操作性，比较容易理解

**缺点：**

1. 该方案没有考虑到人行道的行人影响因素，可能不适合某些路段的实际情况。
2. 模型较为理想化，而真正的红绿灯规则还要考虑具体情况

**模型的改进：**

以遗传算法作为基础对信号灯时长进行模拟，原模型中因为一、二相位信号灯时长相等，三、四相位时长相等所以只将一、三相位的绿灯时长及黄灯时长作为遗传算子，在实际情况中根据实际路段不同也应该考虑二、四相位的信号灯时长。若要提高模型的精确度，应把二、四相位的信号灯时长独立考虑。

# 八、参考文献

[1].姜启源，谢金星，叶俊.数学模型（第四版），北京：高等教育出版社，2011.1

[2].卓金武，李必文，魏永生，秦建.MATLAB在数学建模中的应用，

北京：北京航空航天大学出版社，2014.9

[3].司守奎，孙兆亮，孙玺菁.数学建模算法与应用[M].国防工业出版社，2015

[4].周伟，王秉纲，道路通行能力的理论探讨，交通运输工程学报，

VoI.1.No.1，P92-98，2001.6

[5].多车道公路路段通行能力分析，

P7-11，http://wenku.baidu.com/view/97c5cc380912a216147929a2

[6].苏金明，阮沈勇.MATLAB实用教程，北京：电子工业出版社，2008

[7]. 韩中庚，数学建模竞赛获奖论文精选与点评，北京：科学出版社，2007

[8]. 百度文库，运筹学第十章，

http://doc.mbalib.com/view/5fbb3a21a3e1583c2684b7f967e388bb.html

[9].全国大学生数学建模竞赛组委会，高教社杯全国大学生数学建模竞赛论文格式规范，北京，2009

# 附录

1.建模所使用软件：

写作：Word2013 、Excel2013

编程软件：MATLAB2017a

2.建模代码：

###### 问题一 求解函数

1. %N表示绿灯时间，加速度（m/s）,VM最大速度km/h
2. function [road,count] = fun1(N,a,VM)
3. L=4.8;
4. D=2;
5. T=1;%延迟时间
7. VM =VM/3.6;
8. TM = VM/a;
9. Max= 1000;
10. S=zeros(N,Max);
11. Sn0 = zeros(1,Max);
12. **for** i=1:Max
13. Sn0(i) = (i-1)\*(L+D);
14. end
15. **for** t=1:N
16. **for** n=1:Max
17. TDN = (n-1)\*T;
18. **if** (t<TDN)
19. S(t,n) = Sn0(n);
20. elseif (TDN<=t && t<=TDN+TM)
21. S(t,n)=Sn0(n)-0.5\*a\*(t-TDN)^2;
22. **else**
23. S(t,n)=Sn0(n)-(t-TDN+t-(TDN+TM))\*VM\*0.5;
24. end

27. end
28. end
30. %找到小于0的车辆数
31. count=0;
32. **for** i=1:Max
33. **if** S(N,i) < 0
34. count = count+1;
35. end
36. end
37. count = count\*3;
38. road = S;
39. end

###### 问题二 求解函数

1. %N表示绿灯时间，加速度（m/s）,VM最大速度km/h
2. %Max现在车辆数，DIV表示有多少辆车需要先变道变过来
3. function [road,count] = fun1(N,a,VM,Max,Div)
4. L=4.8;
5. D=2;
6. T=1;%延迟时间
8. VM =VM/3.6;
9. TM = VM/a;
10. %Max= 1000;
11. S=zeros(N,Max);
12. Sn0 = zeros(1,Max);
13. **for** i=1:Max
14. Sn0(i) = (i-1)\*(L+D);
15. end
16. **for** t=1:N
17. **for** n=1:(Max-Div)
18. TDN = (n-1)\*T;
19. **if** (t<TDN)
20. S(t,n) = Sn0(n);
21. elseif  (TDN<=t && t<=TDN+TM)
22. S(t,n)=Sn0(n)-0.5\*a\*(t-TDN)^2;
23. **else**
24. S(t,n)=Sn0(n)-(t-TDN+t-TM)\*VM\*0.5;
25. end
26. end
28. **for** n=(Max-Div):Max
29. TDN = (n-1)\*(T+10);
30. **if** (t<TDN)
31. S(t,n) = Sn0(n);
32. elseif (TDN<=t && t<=TDN+TM)
33. S(t,n)=Sn0(n)-0.5\*a\*(t-TDN)^2;
34. **else**
36. S(t,n)=Sn0(n)-(t-TDN+t-(TDN+TM))\*VM\*0.5;
37. end
38. end
39. end
41. %找到小于0的车辆数
42. count=0;
43. **for** i=1:Max
44. **if** S(N,i) < 0
45. count = count+1;
46. end
47. end
49. road = S;
50. end
51. Ldui = 12;%相当于现在车队为12辆车
52. Q = 400;
53. **for** Q=400:-1:0
54. **new** = round(50/(3600/Q));
55. countZs=[];
56. liuss =[];
57. N2s=[];
58. Nzong=[];
59. InZongs=[];
60. lius=[];
61. Div = round((Ldui+**new**-N2)/2);
62. N1 =Ldui+**new**+Div;
63. N3 = Ldui+**new**+(Ldui+**new**-N2-Div);
64. VM=70;
65. time = 45;
66. N2=5;
67. [road1,count1] = fun2(time,2,VM,N1,Div);
68. [road2,count2] = fun2(time,2,VM,N2,0);
69. [road3,count3] = fun2(time,2,VM,N3,Div);
70. countZ = count1+count2+count3;
71. liu = (count1+count2+count3)/150\*3600;
72. countZs = [countZs;countZ];
73. lius = [lius;liu];
74. N2s=[N2s;N2];
75. Nzong = [Nzong;N1+N2+N3];
76. InZongs=[InZongs;**new**\*3+(105/(3600/Q))\*3];
77. Lv = InZongs./countZ;
78. **if** (Lv<=1)
79. **break**;
80. end
81. end

###### **问题三 遗传算法适应度函数**

（‘’具体遗传函数采用的是matlab自带的GUI工具箱进行实现的）：

1. function [feval] = getValue(x,Q)
2. x1=x(1);
3. y1=x(2);
4. x2=x(3);
5. y2 = x(4);
6. %y2 = 75-(x1+y1)-x2;
7. **if** y2<3
8. feval = 10000;
9. **return**
10. end
11. %  Q=2000;
12. feval=0;
13. %x1,x2 第1,2相象
15. red = x1+y1+2\*(x2+y2);
16. %N表示绿灯时间，加速度（m/s）,VM最大速度km/h
17. [waiteTime1,zuseLv1] = fun3(x1,y1,red,70,Q);
18. feval = feval+waiteTime1\*0.01+zuseLv1;

21. % y2=x(4);
22. red = x2+y2+2\*(x1+y1);
24. %N表示绿灯时间，加速度（m/s）,VM最大速度km/h
25. [waiteTime2,zuseLv2] = fun3(x2,y2,red,40,Q/2);
26. fwaiteTime = waiteTime1+waiteTime2;
27. fzuseLv = zuseLv2+zuseLv1;
28. feval = feval+waiteTime2\*0.01+zuseLv2;
29. **if** feval<0
30. feval=1000;
31. end
32. anw =  [waiteTime1,zuseLv1,waiteTime2,zuseLv2];
33. end

#### 评价各路段函数

1. %N表示绿灯时间，加速度（m/s）,VM最大速度km/h
2. function [waiteTime,zuseLv] = fun3(N,H,red,VM,Q)
3. **new** = round((N+H)/(3600/Q));
4. countZs=[];
5. liuss =[];
6. time = N;
7. N1= round((red)/(3600/Q));
8. N2= round((red)/(3600/Q));
9. N3= round((red)/(3600/Q));
11. [count1,pingjunDenDaiTime1] = fun31(time,2,VM,N1,H+red);
13. waiteTime = pingjunDenDaiTime1;
15. zuseLv = **new**/count1;
16. end
17. %N表示绿灯时间，加速度（m/s）,VM最大速度km/h
18. %Max现在车辆数
19. function [count,pingjunDenDaiTime] = fun31(N,a,VM,Max,RedYtime)
20. L=4.8;
21. D=2;
22. T=1;%延迟时间
23. DenDaiTime=zeros(1,Max);
24. VM =VM/3.6;
25. TM = VM/a;
26. %Max= 1000;
27. S=zeros(N,Max);
28. Sn0 = zeros(1,Max);
29. **for** i=1:Max
30. Sn0(i) = (i-1)\*(L+D);
31. end
32. **for** t=1:N
33. **for** n=1:Max
34. TDN = (n-1)\*T;
35. **if** (t<TDN)
36. S(t,n) = Sn0(n);
37. elseif (TDN<=t && t<=TDN+TM)
38. S(t,n)=Sn0(n)-0.5\*a\*(t-TDN)^2;
39. **else**
41. S(t,n)=Sn0(n)-(t-TDN+t-(TDN+TM))\*VM\*0.5;
42. end
43. **if** (S(t,n)<0 && DenDaiTime(n)==0)
44. DenDaiTime(n)=t;
45. end
46. end
47. end
48. %找到第N秒的时候小于0的车辆数
49. count=0;
50. **for** i=1:Max
51. **if** S(N,i) < 0
52. count = count+1;
53. end
54. end
55. %继续找到其他的等待时间
56. zhouqi = 1;
57. count1 =count;
58. **while**(1)
60. **if** length(find(DenDaiTime==0))==0
61. **break**;
62. end
64. S=zeros(N,Max);
65. Sn0 = zeros(1,Max);
66. **for** i=1:Max
67. Sn0(i) = (i-count1)\*(L+D);
68. end
69. **for** t=1:N
70. **for** n=count1:Max
71. TDN = (n-count1)\*T;
72. **if** (t<TDN)
73. S(t,n) = Sn0(n);
74. elseif  (TDN<=t && t<TDN+TM)
76. S(t,n)=Sn0(n)-0.5\*a\*(t-TDN)^2;
77. **else**
78. S(t,n)=Sn0(n)-(t-TDN+t-TM)\*VM\*0.5;
79. end
81. **if** (S(t,n)<0 && DenDaiTime(n)==0)
82. DenDaiTime(n)=t+zhouqi\*RedYtime;
83. end
84. end
85. end
86. index = count1;
87. **for** i=index:Max
88. **if** S(N,i) <= 0
89. count1 = count1+1;
90. end
91. end
92. zhouqi = zhouqi+1;
94. end
95. pingjunDenDaiTime = sum(DenDaiTime)/Max;
96. end
97. L=4.8;
98. D=2;
99. T=1;%延迟时间
100. a=2;
101. VM =11.1;
102. TM = VM/a;
103. N= 15;
104. Max= 1000;
105. S=zeros(N,Max);
106. Sn0 = zeros(1,Max);
107. **for** i=1:Max
108. Sn0(i) = (i-1)\*(L+D);
109. end
110. **for** t=1:N
111. **for** n=1:Max
112. TDN = (n-1)\*T;
113. **if** (t<TDN)
114. S(t,n) = Sn0(n);
115. elseif (TDN<=t && t<=TM)
116. S(t,n)=Sn0(n)-0.5\*a\*(t-TDN)^2;
117. **else**
118. S(t,n)=Sn0(n)-(t-TDN+t-TM)\*VM\*0.5;
119. end

122. end
123. end
125. %找到小于0的车辆数
126. count=0;
127. **for** i=1:Max
128. **if** S(N,i) < 0
129. count = count+1;
130. end
131. end
132. count = count\*3;