

行车轨迹估计交通信号灯周期问题

摘要

随着城市化进程的加速，城市交通流量日益增加，导致交通拥堵成为城市发展中的一个重大挑战。在这种背景下，准确估计交通信号灯的周期成为提高交通效率和减少拥堵的关键。信号灯周期的优化不仅可以改善车辆的流通效率，减少等待时间，还可以降低汽车尾气排放，从而对环境产生积极影响。

针对问题一，固定周期信号灯周期估计. 对附件 1 中给出的数据，先进行连续性判定，利用前后时刻的 x 、 y 值计算车辆的位移和速度，进而判断车辆是否处于停止状态。考虑到刹车和偶然的停车等影响，采用设置速度阈值和最小停车时长的方式，筛选有效数据。由车辆的停车时间、刹车时间可以得出该车辆在此路口等待的红灯时间，对所有车辆的经过的红灯时间进行计算，可以得出该路口的红灯周期；并且可以根据所有车辆的起步时间进行聚合，起步时间相近的可聚合为一簇，不同簇之间的差值应该正好是信号灯周期或者周期的整数倍，可以算出信号灯的周期。

针对问题二，研究数据不完整、车流量和有定位误差三种因素对于模型计算路口红绿灯时长的精度的影响。首先对于数据不完整和车流量的探讨，进行数据分析和结果提取获得较小比例和不同时间的车辆轨迹数据，代入模型一进行计算，得到信号灯周期的数据，对数据进行分析得出样本数量和红灯持续时间呈正相关。另外，对于定位误差的分析通过改变原始数据的 x, y 坐标值，可以得出，坐标轴的线性变化不会对信号灯周期产生影响。

针对问题三，分析交通信号灯周期的动态变化和检测。首先，需要进行数据准备和预处理。其次，对每个路口的样本车辆轨迹数据进行分析，估计信号灯的周期，并在一定时间间隔内重新估计以检测周期变化，通过设置阈值和使用 CUSUM 方法来判定周期变化点。接着，标识出周期变化的时刻，并计算新旧周期参数。最后，确定识别周期变化的时间和条件，以确保结果的准确性和可靠性。

针对问题四，计算路口方向不确定的信号周期。首先，进行数据的准备和预处理，剔除异常数据和离群值。随后，通过分析车辆的初始位置和结束位置判断车辆行驶轨迹并依据车辆轨迹进行分类，得到 12 类行驶轨迹方向，即可使用问题一的模型进行求解。

关键词：数据清洗，聚类分析，动态周期变化检测，CUSUM 方法

目录

目录	2
一、问题重述	4
1.1 问题背景	4
1.2 待解决的问题	4
二、问题分析	5
三、模型假设	6
四、符号说明	6
五、模型的建立与求解.....	6
5.1 问题一模型的建立与求解	6
5.1.1 模型的建立	6
5.1.2 模型求解	7
5.1.3 模型优化和求解	11
5.2 问题二模型的建立与求解	13
5.2.1 讨论样本车辆比例、车流量和定位误差对模型估计精度的影响.....	13
5.2.2 模型求解	15
5.3 问题三模型的建立与求解	17
5.3.1 模型的建立	17
5.3.2 模型的求解	18
5.4 问题四模型的建立与求解	21
5.4.1 模型的建立	21
5.4.2 数据筛选	22
5.4.3 模型计算	23
六、模型评价与推广	24
6.1 模型的优缺点	24
6.1.1 优点	24
6.1.2 模型的不足	24
6.2 模型推广	25
参考文献.....	25

一、问题重述

1.1 问题背景

交通信号灯是道路交通的基本语言，也是调整平面交叉路，规范道路交通秩序、保障交通的一种道路交通安全设施。特别是随着社会的发展，道路车流量日渐增多，如果可以得到信号灯的变化周期，添加至导航服务中，对交通秩序、安全出行、紧急避难等具有重要意义。但由于许多的交通信号灯并未接入网络，使得直接、省力获取到所有信号灯的数据成为难题。此时，有一家地图服务公司提出或许可以利用大量的行车轨迹数据来估计交通信号灯的周期。

具体来说，这家公司可以通过分析车辆的轨迹数据，识别出车辆在交通信号灯处的行驶状态，从而估计出各个路口信号灯的红绿周期。与此同时，也带来了一些新的技术挑战。

首要的问题是数据的收集和处理。车辆的行驶轨迹是通过样本车辆的 GPS 定位收集的，由于样本车辆和驾驶员的灵敏程度以及 GPS 信号传输的延迟和不稳定性等客观因素，导致数据可能存在误差。因此，需要进行数据的检验，确保分析结果的准确性；其次是行驶状态的判定问题。通过收集的车辆的位移和时间数据来判断车辆的运动状态，然后根据运动状态进行红绿灯周期的估算；最后就是误差因素的考虑，比如样本数据的选择，包括选择通行的路口，测量的时间段等因素对估计时间的影响，因为城市主干道和潮汐车道、早晚高峰期与非高峰期等测到的交通信号灯的周期会有不同。所以需要我们对误差因素的影响程度进行分析，增强说服力。

如果能解决上述问题，不仅可以帮助该公司提升导航服务质量，还能为城市交通管理提供珍贵数据，从而提升城市的交通效率，这将是项非常有意义的工作。

1.2 待解决的问题

基于上述背景以及所获取的数据，构建数学模型拟解决以下问题：

问题一：已知一小时内红绿灯周期不变且路口方向单一，通过给定的车辆 XY-时刻数据估算红绿灯时长。

问题二：讨论少样本、低车流量、定位误差等因素对模型精度的影响。并继续估算附件 2 中的红灯和绿灯的时间。

问题三：信号灯会发生变化，即有可能是周期拉长或缩短，也有可能是周期不变，但红灯和绿灯占周期的比例发生变化。通过附件 3 中的数据修正模型。

问题四：综合轨迹数据分析，使用附件 4 中某路口连续 2 小时内所有方向样本车辆的轨迹数据来识别出该路口信号灯的周期。

二、问题分析

针对问题一：针对附件 1 中给出的数据，先进行连续性判定，数据无误后，进行后续研究。利用前后时刻的 x 、 y 值计算车辆的位移和速度，进而判断车辆是否处于停止状态。考虑到刹车和偶然的停车等影响，采用设置速度阈值和最小停车时长的方式，筛选有效数据。具体来说：先找到车辆速度为 0 的起始时刻和终止时刻，即第一个 0 和最后一个 0，当汽车达到最后一个 0 速度时，意味着车辆开始起步，这意味着红灯刚刚结束。当车速开始从 0 增加时，标记车辆为起步状态。但需特别注意的是，我们判断的红灯时间只包含最后一个 0 之前到阈值速度这一段时间，但并不意味着这就是整个路口的红灯时间。由车辆的停车时间、刹车时间可以得出该车辆在此路口等待的红灯时间，对所有车辆的经过的红灯时间进行计算，可以得出该路口的红灯周期；并且可以根据所有车辆的起步时间进行聚合，起步时间相近的可聚合为一簇，不同簇之间的差值应该正好是信号灯周期或者周期的整数倍，可以算出信号灯的周期。

针对问题二：本题旨在判断数据不完整、有定位误差以及样本数不足这几种因素对于模型计算路口红绿灯时长的精度的影响。首先，需要分析样本车辆比例和车流量，探讨数据不足的影响以及在高峰和非高峰时段数据收集的差异。其次，考虑定位误差对速度估计和停车状态识别的影响。最后，构建模型估算红绿灯周期，并进行数据分析和结果提取，完成表格 2。针对样本不足，采取的方式是先使用全体数据，再对数据做下采样，分析前后结果数据做对比。针对定位误差，采取的是在已有数据上随机叠加 5% 误差，分析前后数据做对比。

针对问题三：本题主要分析交通信号灯周期的动态变化和检测。首先，需要进行数据准备和预处理。其次，对每个路口的样本车辆轨迹数据进行分析，估计信号灯的周期，并在一定时间间隔内重新估计以检测周期变化，通过设置阈值或规则来判断变化的发生。接着，标识出周期变化的时刻，并计算新旧周期参数，填写到

表格中。最后，确定识别周期变化的时间和条件，以确保结果的准确性和可靠性。

针对问题四：问题四要求从附件 4 中的真实路口全体车辆轨迹数据中进行分析，先分类再判断。首先，需要进行数据的准备和预处理，主要取出不完整数据和离群值。随后，通过分析车辆的初始位置和结束位置判断车辆行驶轨迹。通过分析行驶轨迹、时刻和速度情况进行再分类。通过已有确定的几种类型，既可将问题四退化为基本的问题一模型。

三、模型假设

为了方便模型的建立与模型的可行性，这里首先对模型提出一些假设，使得模型更加完备，预测的结果更加合理。

- 1.假设所有驾驶员在见到绿灯的第一时间都会立即起步。
- 2.假设占数据总量 2% 的异常数据不关键，不会对模型产生较大损害。
- 3.假设红绿灯的时长都较为合理，不会出现每三十秒经过一辆车，但红绿灯周期极小，为 15 秒甚至 10 秒的情况，导致车辆间隔是红绿灯周期的整数倍。
- 4.假设车辆行驶过程中没有遇到特殊事故，没有经过终点的车是因为信号消失了而不是整车消失了。

四、符号说明

符号	符号说明
$v_i(t)$	第 i 辆车在 t 时刻的位置
T_{red}	平均的红灯时间
ID	车辆编号
T_b	信号灯周期切换时刻

五、模型的建立与求解

5.1 问题一模型的建立与求解

5.1.1 模型的建立

- (1) 速度表达式：

设 $x(t)$ 和 $y(t)$ 分别是时间 t 的函数，表示物体在二维空间中的 x 和 y 坐标。

瞬时速度可以用向量 $v(t)$ 表示包含 x 和 y 方向的速度分量。

$$V(t) = (v_x(t), v_y(t)) \quad (\text{式 5-1})$$

$$\text{其中, } v_x(t) = \frac{dx}{dt} \quad v_y(t) = \frac{dy}{dt};$$

如果只有离散的数据点而不是连续函数，可以使用差分方法来近似导数，即使用相邻点之间的变化率来估计瞬时速度。假设有时间点 t_1 和 t_2 ，以及这两个时间点对应的坐标 (x_1, y_1) 和 (x_2, y_2) ，那么在 t_1 和 t_2 之间的平均速度向量可以近似为：

$$v_{avg} = \left(\frac{x_2 - x_1}{t_2 - t_1}, \frac{y_2 - y_1}{t_2 - t_1} \right) \quad (\text{式 5-2})$$

设第 i 辆车在 t 时刻的位置为 $(x_i(t), y_i(t))$ 则其速度为：

$$v_i(t) = \sqrt{(x_i(t + \Delta t) - x_i(t))^2 + (y_i(t + \Delta t) - y_i(t))^2} / \Delta t \quad (\text{式 5-3})$$

(2) K-means 聚类

概念：将样本划分为由类似的对象组成的多个类的过程。聚类后，我们可以更加准确的在每个类中单独使用统计模型进行估计、分析或预测；也可以探究不同类之间的相关性和差异性。

流程：指定需要划分的簇的个数或者是划分条件；选择初始的聚类中心；根据初始聚类中心和划分条件进行聚类；重复以上步骤，检查中心是否收敛。

欧式距离：对于 $x (x_1, x_2, \dots, x_n)$ 点坐标和 $y (y_1, y_2, \dots, y_n)$ 点坐标，两者之间的欧式距离为

$$\begin{aligned} d(x, y) &= \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2} \\ &= \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \end{aligned} \quad (\text{式 5-4})$$

5.1.2 模型求解

(1) 数据分析

利用 MATLAB，对给出的附件 1 中 A1、A2、A3、A4、A5 五个车辆轨迹数据进行异常值和缺失值的判定。根据对车辆轨迹的分析，如下图 1 所示，发现所有数据均为连续数据且均无异常，可以进行后续分析。

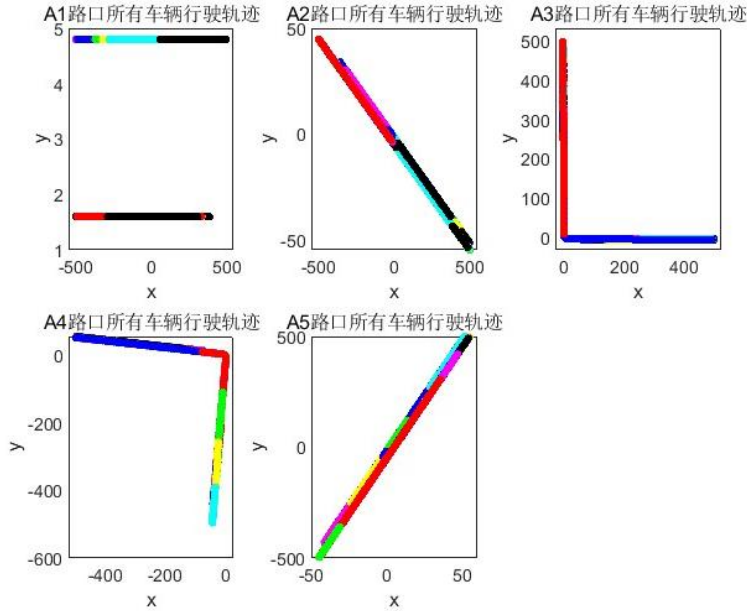


图 1 所有车辆行驶轨迹图

为了更加直观的展示运动轨迹，以 A1 中 ID18, ID32, ID54 为例，绘制了其 x 轴，y 轴的运动轨迹。可以从图像中看出车辆的移动和行驶方向，其中 x-t 图像中曲线的斜率代表的是车辆的行驶速度。

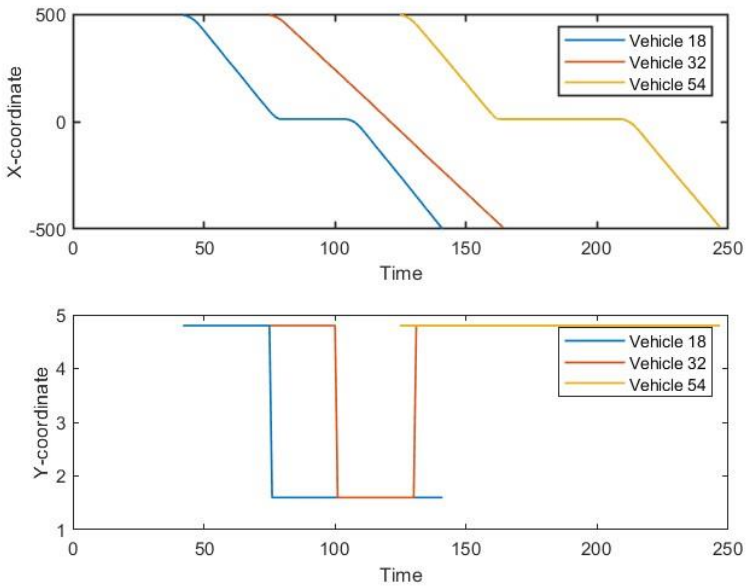


图 2 对 x-y 轴位置的分析

(二) 求解红灯时间

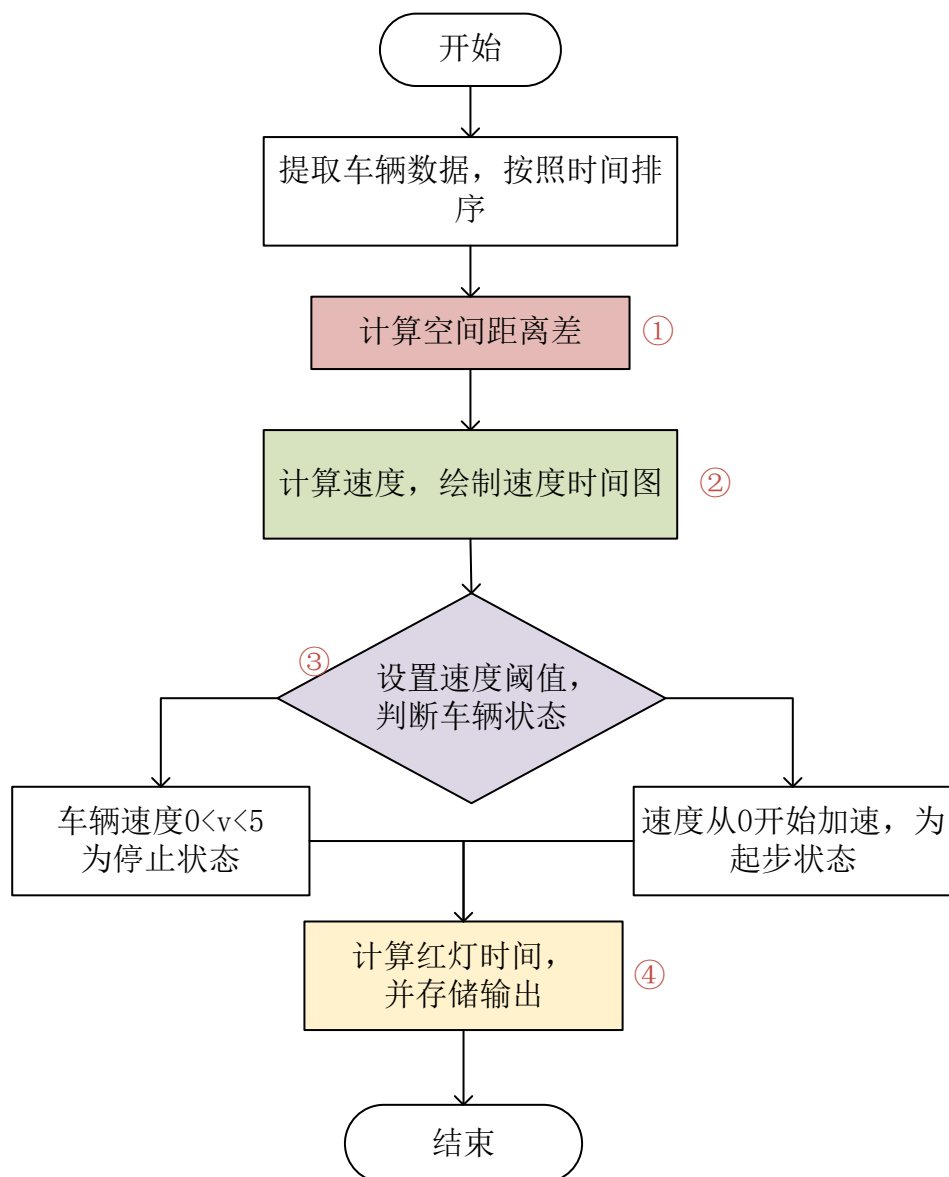


图 3 求解红灯时间的步骤

① 计算空间距离差。提取表格中的 x, y 坐标和时间, 令

$t = \text{vehicle_data.time};$

$x = \text{vehicle_data.x};$

$y = \text{vehicle_data.y};$

利用 `diff` 函数求解空间上的距离差, 如令 $d_x = \text{diff}(x);$

② 计算车辆速度, 绘制速度-时间图像。通过对车辆在连续时刻的位置变化情

况进行观察，我们可以使用时间与位置的微分来计算车辆的瞬时速度。速度的变化能够反映出车辆是否遭遇到了交通信号或拥堵。对于每个时间片段,按照前面的定义计算车辆的速度 $v_i(t)$ 。以 A1 位列，选取 ID18、ID32、ID52 在路口速度变化和停车状态。从图中可以观察到车辆的速度变化以及减速、加速和停止的时刻。

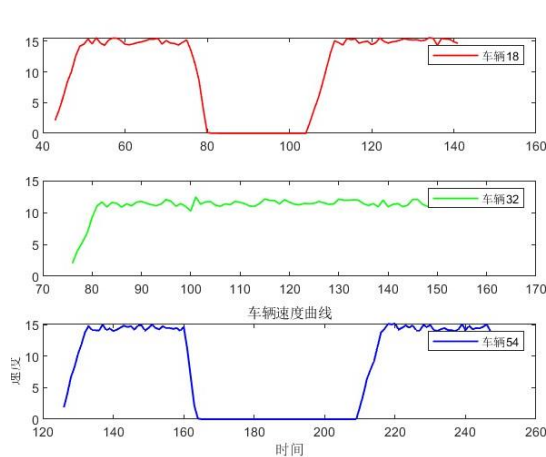


图 4 车辆速度示例

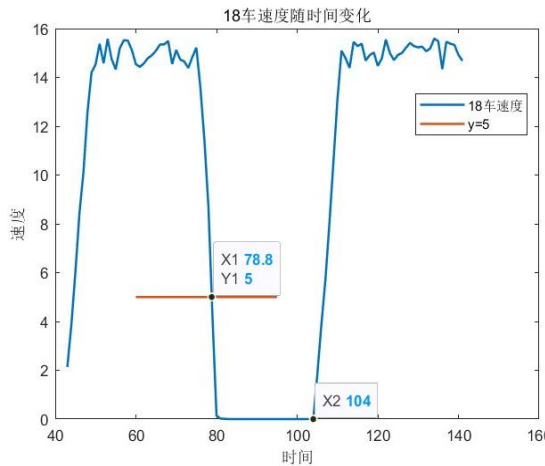


图 5 ID18 速度随时间的变化

③ 设置阈值，定义车辆速度状态。考虑在红绿灯交替时,车辆速度会出现明显的变化。对于每个信号灯可能的跳变时刻 t_j ,可以观察其前后车辆状态分布的变化情况，我们基于速度变化**设定阈值为 5m/s**（因为根据速度数据可以看出，车辆在完全停下之前会有一段持续 3 到 5 秒的减速时间，这段时间显然是因为驾驶员看到了红灯并且采取制动措施所形成的，因此需要对速度设置一个合理阈值,用于补偿这一部分时间，本次建模大部分时候都将车速阈值设置为 5m/s 时），当速度小于这个阈值将它看作停止状态；反之，如果主要是从停止状态转为加速状态，那么可能是绿灯开始。当遇到红灯时，速度会逐渐降低直至为 0，而遇到绿灯时,速度会从 0 开始逐渐升高，定义此时属于起步状态。对于每个停止事件，找到相应车辆ID且时间大于该停止时间的最近的启动时间将相邻的同种标记(红灯或绿灯)的切换时刻进行匹配，计算它们之间的时间间隔，即可得到对应状态的持续时长。以 A1 中 ID18 为例：

从图 5 中可以观察到：点 (x_1, y_2) 表示在 78.8 秒时，ID18 的速度已经小于 5 了，即我们认为此时车辆的状态为停止状态，在点 x_2 处，即时间 104 秒后，车辆处于起步状态，此时红灯时间的计算为两点之间的时间差 $x_2 - x_1 = 35.2$ 秒。但

是目前的分析是基于单个停车时长的统计，需要考虑时间窗口内的多个停车和启动事件，才能够准确地反映周期性。

④为了得到精确的红绿灯周期估计，我们需要对更多车辆或者更长时间的数据进行类似的分析，以 A1 路口为例，我们可以得出：

最短的停车时间为 1 秒，最长的停车时间为 106 秒，我们通过计算多个停车事件的持续时间来设置算法估计红灯的平均时长 T_{red} 。如下图所示，通过计算，得出了 A1 中所有车辆的等待红灯时长，经计算后，得出 A1 路口的 T_{red} 约为 45 秒。

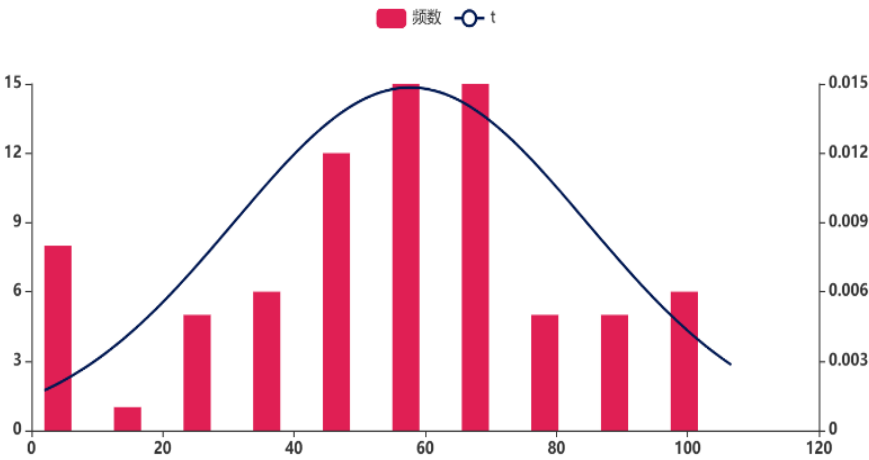


图 6 A1 路口所有车辆的等待红灯时长

5.1.3 模型优化和求解

(1) 优化后求解红灯时间

通过对上图的分析，我们可以看出车辆的停车分布看起来是很不均匀的，他显示了几个频率较多的停车时间，有些停车时间太短，有些的停车时间过长，根据文献调研和现实考虑，应对该部分数据进行删除。同时也对模型进行修正，包括停车阈值：检查用于判断车辆是否停止的速度阈值是否合理，过大或过小的阈值可能导致停车状态的误判；停车状态的连续性：检查是否正确地标记了连续的停车状态，可能存在停车状态被间断识别、多次标记的情况。

另外，根据常识可知，即使司机很远就看到了红灯，但由于车辆距离路口还非常远，那么大部分司机都会选择继续滑行一段时间之后再刹车。或者是路口已经是红灯了，但由于司机达到路口的时间较晚，因此估算的红灯时长小于实际红灯时长。假设一个红灯持续 70 秒，那么绝大多数车辆的等待时间都远小于 70

秒，只有上一个绿灯周期最后一辆没能通过路口的车才需要等够 70 秒。对于估算出的红灯时长应当结合实际经验，尽量选取偏大的估算值，而不是忽略实际问题，采取数学方法取平均、做加权或是正态分布拟合。

经过对模型进行修正，我们重新得到了修正后的结果：

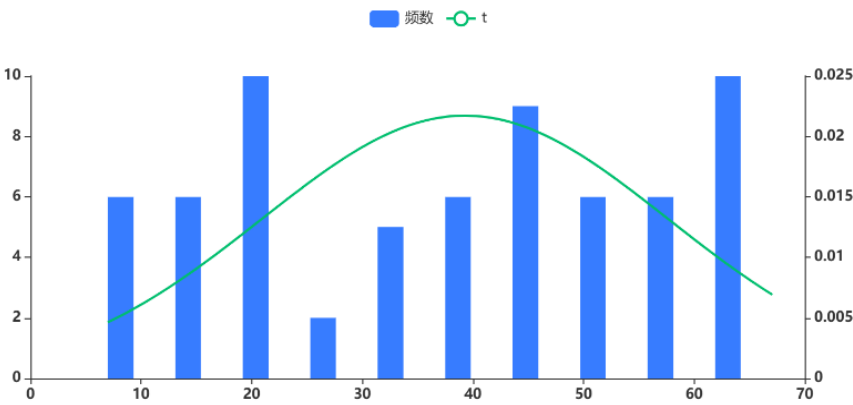


图 7 优化后 A1 路口所有车辆的等待红灯时长

通过计算，得出了 A1 中所有车辆的等待红灯时长，经模型优化后，选取等待红灯的最大值估计得出 A1 路口的红灯时长为 68 秒。

也可以利用箱线图对比优化前后的效果。箱线图(Box-plot)又称为盒须图、，是一种用作显示一组数据分散情况资料的统计图，要用于反映原始数据分布的特征，还可以进行多组数据分布特征的比较因形状如箱子而得名，在各种领域也经常被使用，常见于品质管理。从图中我们发现优化前后的最大、最小值间隔很大，较为分散，可能存在一些异常值，经过优化后，可以使数据更加集中，精准。

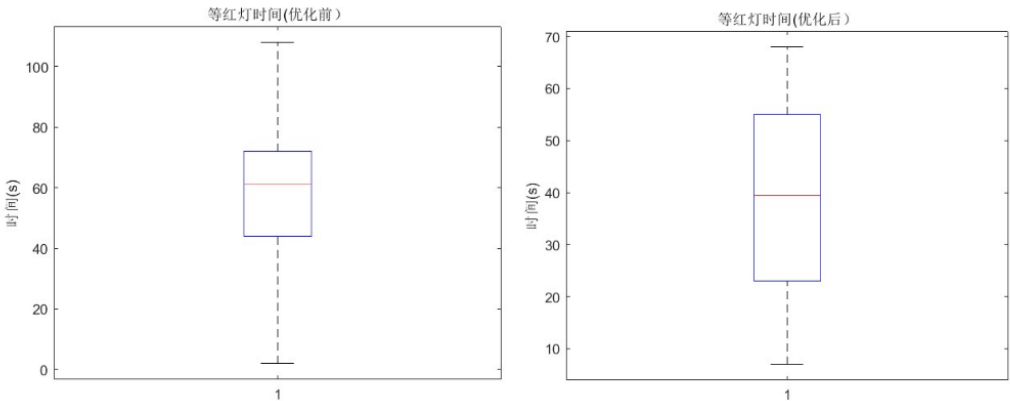


图 8 A1 路口优化前后红灯时间的对比

同理，用相同的方式求出其他 A2、A3、A4、A5 路口的红灯时长。

(2) 求解周期时间

以 A1 为例，求出路口车辆的起步时刻，如图 8，根据所有车辆的起步时间进行一定的聚合，起步时间相近的可分为一簇，相邻两簇的起步时间的差值应该正好是信号灯周期，由此可以算出信号灯的周期如图 9。在本题中，我们选取起步时间在 20 秒以内的为一簇，两簇之间的差值即为信号灯的周期。经计算，可得出 A1 路口的信号灯周期为 105 秒。

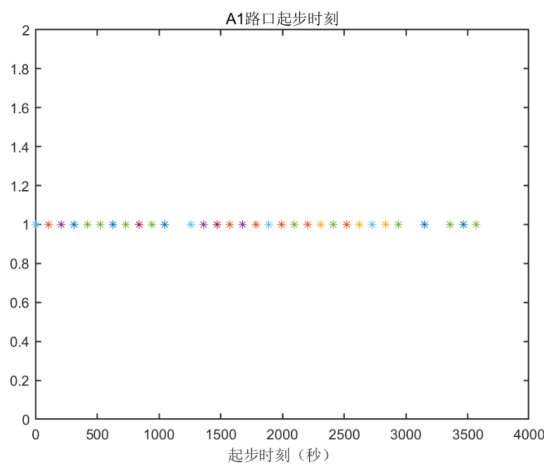


图 9 A1 路口车辆的起步时刻

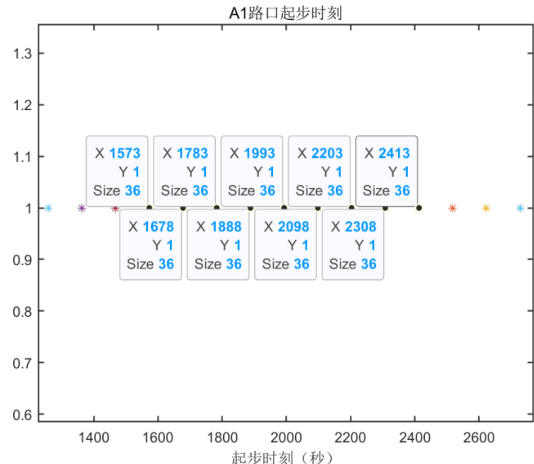


图 10 A1 路口信号灯的周期间隔

(3) 求解绿灯时间

根据交叉口的信号设计，红灯时长和绿灯时长通常是固定的，或者是按照特定的周期模式变化的。因此，一旦我们有了估计的红灯周期和完整周期，我们也就能够相应地推算出绿灯的持续时间。

(4) 具体结果如下

表 1：路口 A1-A5 各自一个方向信号灯周期识别结果

路口	A1	A2	A3	A4	A5
周期	105	88	105	88	88
红灯时间	68	52	71	68	57
绿灯时间	37	36	34	20	31

5.2 问题二模型的建立与求解

5.2.1 讨论样本车辆比例、车流量和定位误差对模型估计精度的影响

样本车辆比例：主要是分析不同的车辆比例对于这个信号灯绿灯持续的影

响。通过筛选样本车辆编号来改变样本比例，以B1路口1小时内的数据为例，在原有73辆车的基础上，随机选取36、18辆车的红绿灯数据（保证这些车辆位于不同的时间段）来进行对比。通常来说当数据量减少时，估计的准确性和稳定性可能会降低；而当数据量足够时，估计的结果可能会趋于稳定，具体结果如下：红灯时长估计：

表 2：不同样本比例的红灯时长

样本比例	100%	50%	25%
B1（红灯时长）	77	77	77
B5（红灯时长）	82	80	80

从这两次计算中发现，随着样本车辆比例的减少，平均红灯持续时间基本保持不变，这意味着我们估算的红灯的平均持续时间比较准确，对样本具有包容性。绿灯时长估计：

表 3：不同样本比例的绿灯时长

样本比例	100%	50%	25%
B1（绿灯时长）	28	27	27
B5（绿灯时长）	34	152	152

与红灯时长相反的是，绿灯的平均时长随样本率减少而有了显著的变化。从图中可以看出 B1 路口的绿灯时长基本不变，但 B5 路口的绿灯时长出现了明显的异常值。这表明数据的质量和有效性是不同的，较小的样本比例会影响模型对于周期的计算，从而导致了不合理的过长的绿灯时间。

车流量：车流量的增加会导致样本车辆通过路口的时间变化，从而影响模型的精度。通过筛选不同的时间段来改变车流量数据。在原有1小时数据的基础上，选择三个不同时间段的红绿灯数据与全样本的红绿灯数据进行对比。

红灯时长估计：

表 4：不同车流量的红灯时长

样本时间	0-20 分钟	20-40 分钟	40-60 分钟	0-60 分钟
B1（红灯时长）	77（29 辆）	73（31 辆）	73（17 辆）	77（77 辆）
B5（红灯时长）	80（23 辆）	72（11 辆）	75（19 辆）	82（53 辆）

从这两次在此次计算中，可以发现，当车流量接近的时刻，等待红灯的时间也是接近的。

绿灯时长估计：

表 5: 不同车流量的绿灯时长

样本时间	0-20 分钟	20-40 分钟	40-60 分钟	0-60 分钟
B1（绿灯时长）	28（29 辆）	32（31 辆）	29（17 辆）	28（80 辆）
B5（绿灯时长）	34（23 辆）	37（11 辆）	41（19 辆）	34（53 辆）

与红灯时长估计类似，当车流量接近的时刻，等待绿灯的时间也是接近的。

定位误差：定位误差会导致实际通过红绿灯的时间与轨迹数据中记录的时间不一致，从而影响数据的准确性，进而影响模型的精度。通过在原始数据的坐标上增加一个偏移量（5%），来模拟外界因素引起的波动。

表 6: 不同 x,y 坐标的红绿灯时长

B3	$x+5\%*x$	$y+5\%*y$	$x+5\%*x \& y+5\%*y$	$x-5\%*x \& y-5\%*y$
红灯	53	53	53	53
绿灯	34	34	34	34



图 11 可视化定位误差前后B3路口的红绿灯时长

通过对比，可以看到上述表格中 x 坐标和 y 坐标改变对模型计算信号灯的红绿灯时长并没有影响。这是因为线性的定位误差导致的位置变化并不会改变车辆的速度，而我们的模型正是通过判断车辆到达阈值速度的时刻来求解红灯时长，速度不变红灯时长自然也不变。

5.2.2 模型求解

利用 MATLAB，对给出的附件 2 中 B1、B2、B3、B4、B5 五个车辆轨迹数据进

行异常值和缺失值的判定，并对这类异常值进行舍去。根据对车辆轨迹的分析发现：所有数据均为连续数据且均无异常，可以进行后续分析。

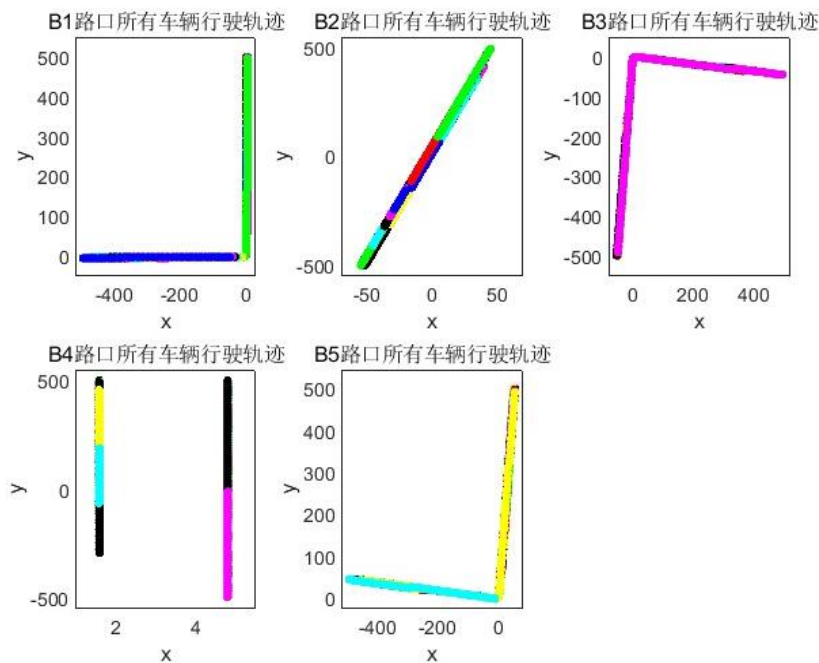


图 12 所有车辆的行驶轨迹

利用问题一中建立的模型，可以计算红灯时长，将模型依次导入 B1、B2、B3、B4 和 B5 数据集合中进行估算，即可得到结果。同时，针对起步时间进行聚类分析，相邻两簇之间的起步时间差为信号灯的周期，从下图可以看出，以数据 B1 为例，得出的起步周期基本保持一致，说明该计算模型具有一定的正确性。

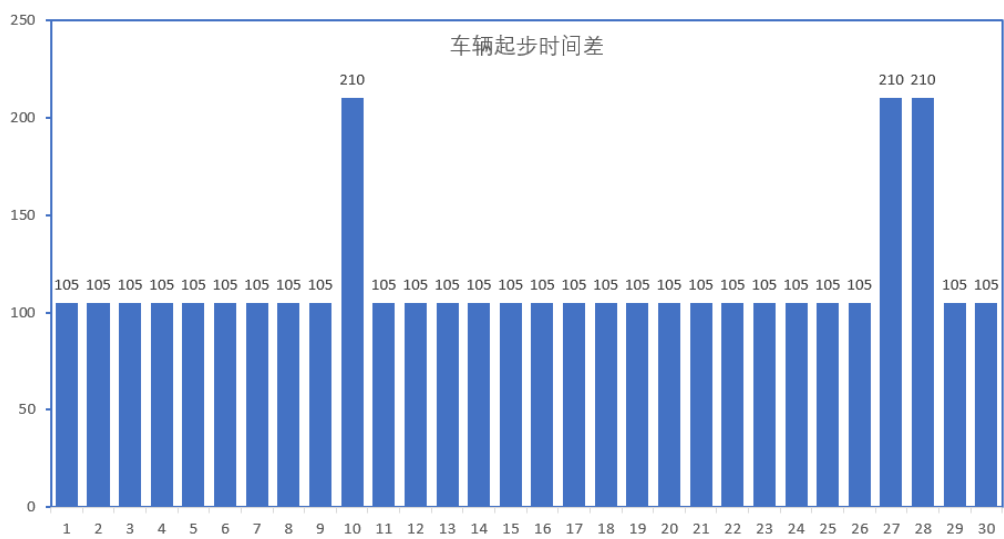


图 13 B1 路口相邻两簇车辆的起步时间间隔

表 7: 路口 B1-B5 各自一个方向信号灯周期识别结果

路口	B1	B2	B3	B4	B5
周期	105	116	87	104	116
红灯时间	77	82	53	77	80
绿灯时间	28	34	34	27	36

5.3 问题三模型的建立与求解

5.3.1 模型的建立

首先，需要进行数据准备和预处理。其次，对每个路口的样本车辆轨迹数据进行分析，估计信号灯的周期，与之前的模型大致相同，特别是，在此处需要计算得出筛选后的有效停车持续时间。

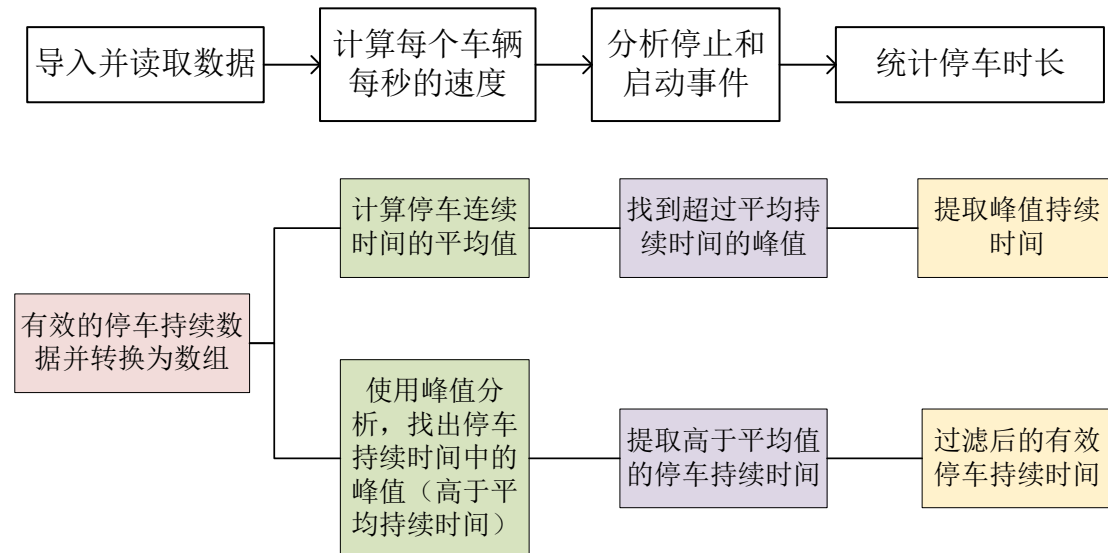


图 14 有效停车持续时间筛选流程

接着，一定时间间隔内重新估计以检测周期变化，通过设置阈值或规则来判断变化的发生。接着，标识出周期变化的时刻，并计算新旧周期参数，此时我们使用 CUSUM（Cumulative Sum Control Chart）方法进行信号灯周期变化的检测。

CUSUM 是一种有效的统计技术，它主要用于监控和识别过程或系统中的变化点，此方法非常适合于交通数据，特别是在需要识别交通信号灯周期变化或其他系统性变化的情况下。CUSUM 是一种序列化的检测方法，它通过计算数据点与过程平均值或目标值的累积偏差来确定一系列数据点中是否存在显著的变化。对本题来说相当于，信号灯的红绿灯周期改变，CUSUM 值将会显著偏离零，表

现出持续的上升或下降趋势，由此达到检测的目的。

使用 CUSUM 方法，首先需要确定目标值和阈值：本题中，我们的目标值是交通信号灯的平均红灯时长；阈值是确定何时认为系统发生变化的参数，需要根据具体数据和需求进行调整。其次需要计算 CUSUM 分数：选择一个起始点，通常基于数据集的平均值或中位数，指代本题中的有效停车持续时间，对于每一个新的数据点，计算其与目标值的偏差，然后将这个偏差累加到之前的 CUSUM 分数上。分别计算向上和向下的 CUSUM，以便能够检测出任何方向上的变化。当 CUSUM 分数超过预设的阈值时，可以认为是发现了一个变化点。

一旦检测到变化点，分析变化前后的数据，以评估变化的性质和影响。

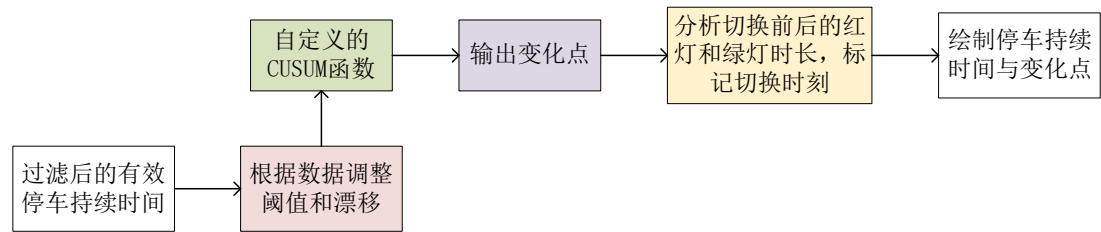


图 15 动态周期变化检测流程

5.3.2 模型的求解

以 C1 为例进行说明，根据之前的流程计算有效的停车持续时间数据：

[3 38 56 44 1 39 50 23 1 7 6 2 17 41 37 8 26 20 4 30 3 9 16 41 34 40 23 8 41 21
7 8 22 13 10 15 1 13 46 24 3 13 28 18 1 12 45 41 14 6 10 40 7 2 42 19 3 9 8 19 4 7 13
11 8 6 35 20 13 11 8 10 9 16 32 22 28 16 28 17 14 1 38 22 1 4 1 42 19 20 42 12 5 19
17 15 34 11 19 5 10 9 12 32 30 7 8 45 22 4 38 37 26 15]

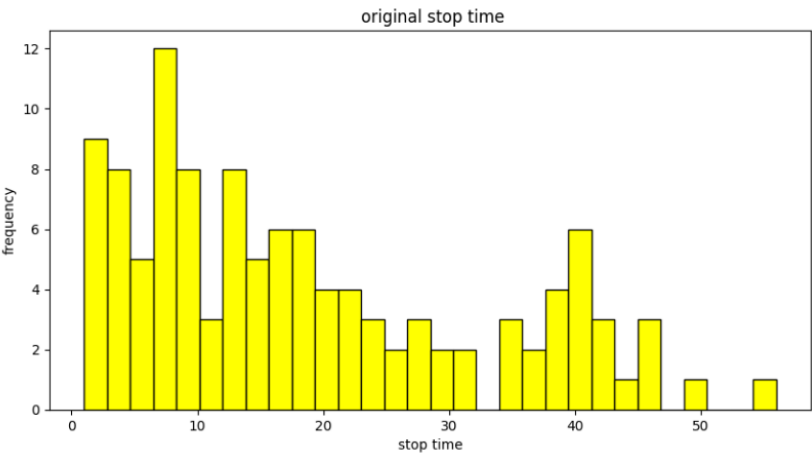


图 16 原样本数据计算的有效的停车持续时间

可以使用峰值分析确定停车持续时间中的主要峰值，将停车持续时间大于平均值的的数据视为有效数据，低于平均值的视为异常值进行筛选。

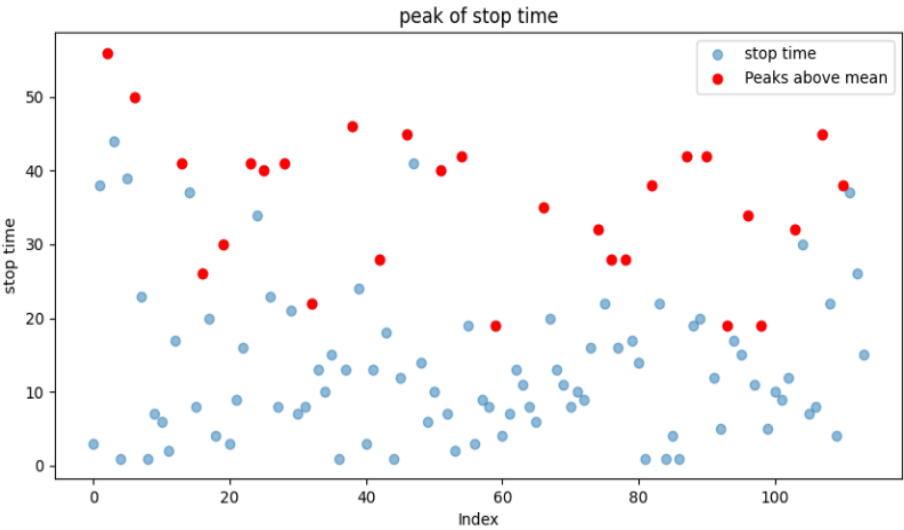


图 17 峰值分析检测

对峰值设定阈值外的值进行剔除处理，得到过滤后有效停车持续时间

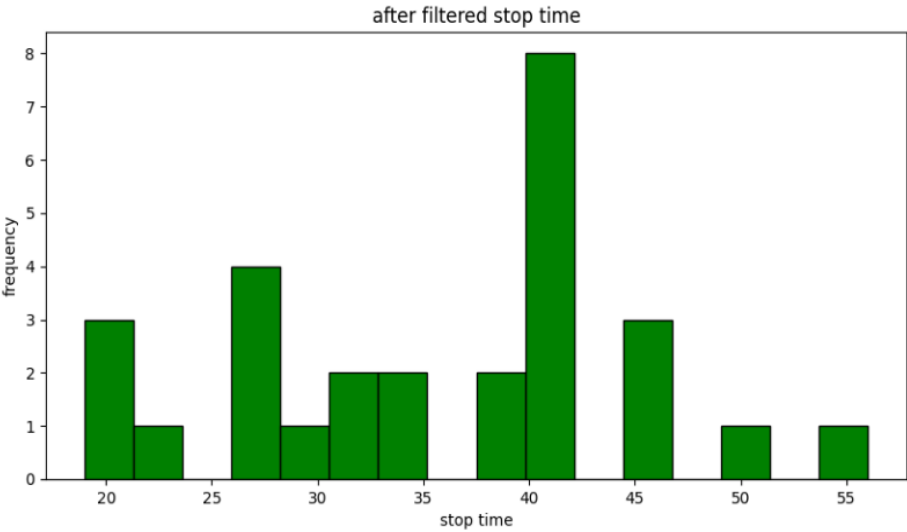


图 18 过滤后的有效停车持续时间

输出得到，过滤后的有效停车持续时间：

[56, 50, 41, 26, 30, 41, 40, 41, 22, 46, 28, 45, 40, 42, 19, 35, 32, 28, 28, 38, 42, 42, 19 34, 19, 32, 45, 38]

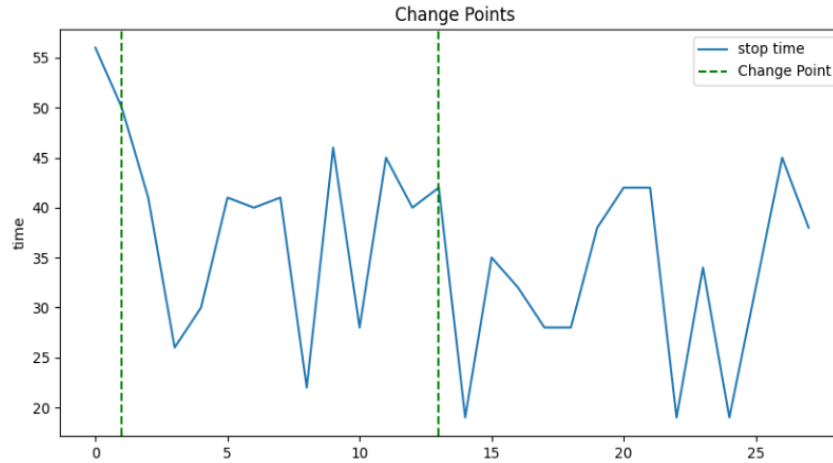


图 19 周期变化点

最终结果： 周期变化点： 1、13

T_b 切换时刻: 226 秒

切换前红灯平均时长: 56 秒,

切换后红灯平均时长: 35 秒, 绿灯平均时长: 62 秒

T_b 切换时刻: 839 秒

切换前红灯平均时长: 39 秒, 绿灯平均时长: 43 秒

切换后红灯平均时长: 33 秒, 绿灯平均时长: 63 秒

表 7： 路口 C1-C6 各自一个方向信号灯动态周期识别结果

路口	C1	C2	C3	C4	C5	C6
周期 1 红灯时长 (秒)	56	48	54	66	33	55
周期 1 绿灯时长 (秒)		38	62		117	35
周期切换时刻	226	246	164	87	1661	96
周期 2 红灯时长 (秒)	35	54	59	60	39	54
周期 2 绿灯时长 (秒)	62	100	45	53	88	94
周期切换时刻	839	1770	546	118	2073	417
周期 3 红灯时长 (秒)	33	59	59	69	35	61
周期 3 绿灯时长 (秒)	63	96	44	24	123	17
周期切换时刻	无	无	593	无	无	无
周期 4 红灯时长 (秒)			58			
周期 4 绿灯时长 (秒)			52			

5.4 问题四模型的建立与求解

5.4.1 模型的建立

问题四要求对路口两小时内所有车辆数据进行分析,判断各路口红绿灯的情况。主要包括红灯、绿灯的持续时间和循环周期,以及可能存在的信号等周期变化的问题和直行左转混行的问题。因此,本研究将通过以下几步对问题进行建模分析:

步骤一:剔除不完整数据与离群数据,防止对结果产生不良影响。

步骤二:通过初始位置与运动方向对轨迹进行分类。

步骤三:模型退化为求解十二个方向模型的解的问题。

(1) 异常变量的剔除

为了更加直观的观察,展示所有车辆路过路口所构成的轨迹图,如下图所示

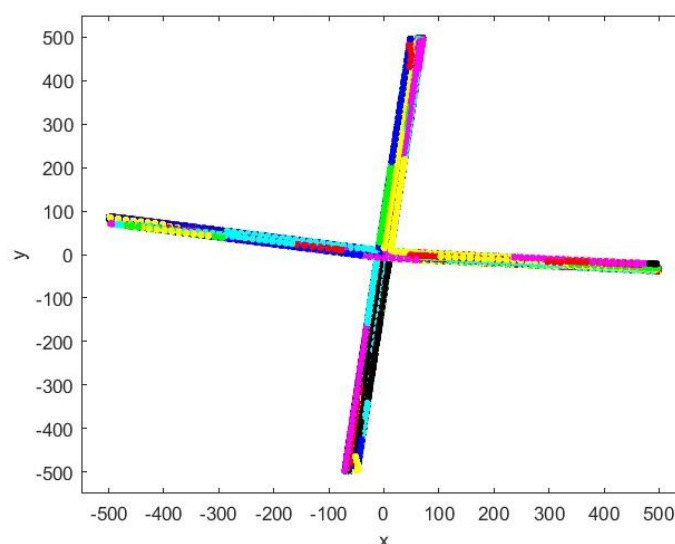


图 20 该路口所有车辆行驶轨迹

经过对轨迹的分析,我们发现由于数据录入不完整或干扰等问题,造成了如 5468 号车和 5467 号车分别只有一条和两条记录,也就是说两台车只被记录了三秒钟的时间。由于估算速度时采用了差分的方式,因此会对结果产生影响。由于干扰等问题,部分车辆会在接近路口时变道或者刹车之后再启动的问题,这类数据需要通过预测去消去。受影响规模约 30 台。从轨迹图观察还能再排除一些离群值,总数约占全部样本的 2% (59/3298 条)。

(2) 聚类分析

通过位置信息判断出发点和运动方向,进而对车辆的运动进行聚类分析,用

于后续分析红绿灯的工作情况。首先，提取 t 时间内，某台车第一次出现的时刻和最后一个时刻，即出发位置和最终运动方向。观察每辆车的第一个时刻可以发现，所有车辆只会从四个确定的坐标位置出发 ($x = 495.39$ $y = -21.7$ 、 $x = 48.27$ $y = 496.27$ 、 $x = -496.22$ $y = 71.29$ 和 $x = -48.27$ $y = -496.27$ 。) 通过车辆可能进行的左转、直行和右转可知，车辆总共有 12 种可能的方向进行。

(3) 估算车辆的运动轨迹并分类

在进行合理分类之后，可通过计算连续坐标点之间的变化，估计车辆的行驶方向。例如，如果 x 坐标随时间增加而 y 坐标减少，车辆可能是向东北方向行驶。

5.4.2 数据筛选

通过对车辆起始位置和位移的分析，可以发现发现：所有的车辆都是从 A、B、C 和 D 四个初始位置出发去往其他三个点。但数据中也发现有些不完整数据，如下右图所示，红色圆内的部分数据是从初始 ABCD 位置出发，但并未到达终点位置而是消失在了圆圈中，这些样本点可能通过了路口也可能没有通过路口，因此对于这类数据我们选择筛去，一个重要原因是经过严密计算，筛去的数据占原数据集的比例小于 2%，筛去后不影响参数的估计。

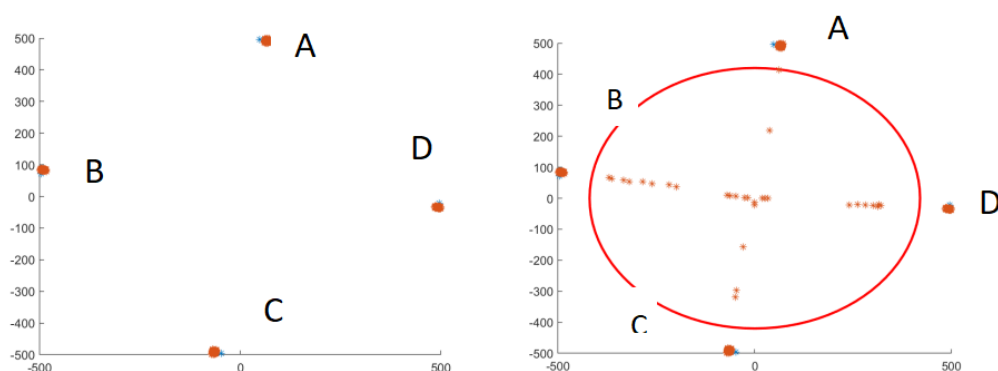


图 21 路口所有车辆起始位置

对于筛选后的数据，根据上图的示意可以发现，我们可以将它们进一步拆分为 12 种情况，即源码中的 case1 - case12。在示意图上即可表示为：{AB、AC、AD}；{BA、BC、BD}；{CA、CB、CD}；{DA、DB、DC} 这几组方向。这点也可从交叉路口的放大图上更显然的发现。此处需要说明的是：AC 和 CA 是不等价的，一个代表自北向南行进，另一个代表自南向北行进。据此我们可以将数据完全分类，每组数据对应一个路口的一个行驶方向，对于每一组数据来说

可以延续使用问题一的模型，进行红绿灯周期的计算。

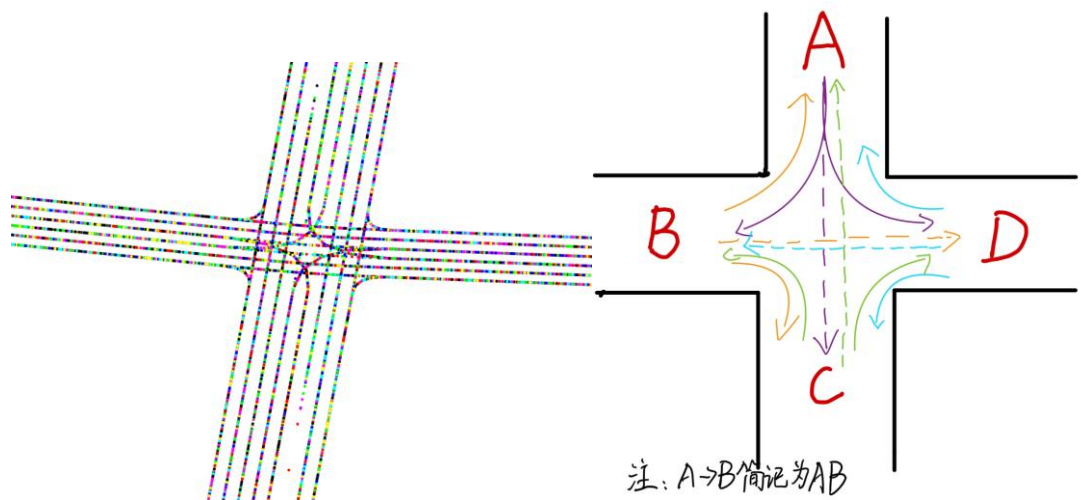


图 22 路口车辆的行驶方向示意图

5.4.3 模型计算

利用问题一中建立的模型，可以计算对应的红灯时长和绿灯时长，导入 12 个方向的数据进行估算可得到如下结果。

表 8： D 路口 不同方向信号灯周期结果

路口/方向	周期	红灯	绿灯
AB	142	95	47
AC	141	97	44
AD	141	127	14
BA	139	100	39
BC	138	113	25
BD	141	103	38
CA	142	98	44
CB	141	98	43
CD	138	118	20
DA	140	101	39
DB	142	105	37
DC	139	117	22

六、模型评价与推广

6.1 模型的优缺点

6.1.1 优点

1、将直接求解红灯持续时长和绿灯持续时长转变为：求红灯时长和红绿灯周期的问题。周期的求解有具有较高的稳定性，几乎不受样本抽样的影响。因此准确性相对较高。

2、由于对运动轨迹进行了差分，一方面相当于对速度进行了分析，另一方面也可以看作是对相对位置进行了分析。在定位数据发生“漂移”等线性变化的情况时，依然能很好的用于估计红绿灯时长。

3、采用了统计直方图的方式分析了路口红绿灯变化的周期，红灯持续时长会在直方图中反应，最后用过直方图便能较为准确的确定红绿灯有几类周期，再结合数据推算该周期内红灯和绿灯的持续时间。

4、通过数据清洗除去了残次数据，进一步完善了数据的完整性。通过初步分析准确估计了车辆的 12 种运行方式，即四个路口的直行、左转和右转。又依据这些信息，将整体数据划分为了 12 组，每一个组或称簇都只含有一种情况的数据。于是原本复杂的问题即可退化为与第一题几乎相似的信号灯估计的问题。

5、算法的空间复杂度控制的较好，基本保持在了 $O(n)$ 。通过确定红绿灯的具体位置的方式，有望将空间复杂度优化到 $O(1)$ 。

6.1.2 模型的不足

1、由于存在程序健壮性的问题，并不能一套代码通用于所有 A1-A5、B1-B5、C1-C6 和 D 组的数据，部分时候会出现估计的红灯时间异常长的情况，此时需要针对性的做少量优化。造成了程序代码的冗余与繁杂。

2、在定位数据漂移对模型精度的估计过程中，加入的轨迹噪声是固定偏移 +5%，这显然与实际情况不符。实际的位置偏移是非线性的，主要取决于云层树木遮挡和高楼遮挡，使得信道产生多径衰落，此时会导致获取卫星定位的时间比一般时期更长，于是会产生更大的非线性偏移。由于考虑了位置的线性偏移并未考虑到模型产生的非线性偏移，因此对模型的准确性会有盲目乐观的不良后果。

3、在第三题，分析红绿灯周期是否有变化的过程中，通过直方图估计并判

断周期的方法不够合理，存在经验性判断的嫌疑，较难从数学上给出证明。只能说是符合直觉。但完备性不足。

4、通过放大十字路口车辆运动构成的轨迹，可以明显看出，整个路口的马路有：南北走向该方向为双向八车道，东西走向该方向双向六车道。并且含有直行右转合用道四条，直行左转合用道两条，但在模型中只是简单的考虑了 12 种基本情况，并未含合用道的情况。这将意味着，左转直行、右转直行两类共八中情况没用被纳入考虑，人为的进行“解耦”。这也会对模型的实际含义产生影响。

5、该模型的算法时间复杂度极高，初步估计在 $O(n^2)$ 到 $O(n^3)$ 之间，对于第四题，算法复杂度肯定超过了 $O(n^3)$ 。

6.2 模型推广

事实上，目前高德软件有限公司已就《红绿灯周期时长的挖掘方法》申请并获批了国家发明专利且已大规模应用。但类似的分析手法、处理手段完全可以应用于非开放环境的路径优化等问题。

参考文献

- [1] 王志建, 金晨辉, 龙顺忠, 等. 基于轨迹数据的信号交叉口排队长度估计[J]. 科学技术与工程, 2022, 22(21): 9407-9143. Wang Zhijian, Jin Chenhui, Long Shunzhong, et al. Queue length of signal intersection based on trajectory data[J]. Science Technology and Engineering, 2022, 22(21): 9407-9143.
- [2] 王雷, 安实, 杨海强, 等. 基于出租车轨迹数据的交通异常识别算法[J]. 科学技术与工程, 2018, 18(32): 239-247. Wang Lei, An Shi, Yang Haiqiang, et al. Traffic anomaly recognition algorithm based on taxi trajectory data[J]. Science Technology and Engineering, 2018, 18(32): 239-247.
- [3] 吴庆哲, 周俊. 基于模糊控制的多路口路段交通信号灯信号周期的研究[J]. 工业控制计算机, 2019, 32(6).
- [4] 郭炜杰, 包晓安. 单交叉路口交通灯实时配时算法的研究[J]. 工业控制计算机, 2012, 25(3): 34-36.
- [5] 梁泉, 翁剑成, 胡娟娟, 等. 多源数据驱动下公共交通出行行为特征挖掘与预测[J]. 科学技术与工程, 2021, 21 (28): 11921-11931.

附录

第一问：计算 A1 路口的红绿灯时间

Q1-1:数据读取并可视化

```
clear
%%加载数据
%%绘制路口所有车辆轨迹
data1=readtable('A1.csv');
subplot(2,3,1)
gscatter(data1.x,data1.y,data1.vehicle_id,'brgyckm','.',10);
xlabel('x');
ylabel('y');
title('A1 路口所有车辆行驶轨迹');
%%绘制 A1 路口车辆编号为 18, 32, 54 的坐标变化图
% 提取车辆编号为 18, 32, 54 的数据
vehicle_ids=[18,32,54];
selected_data=data1(ismember(data1.vehicle_id,vehicle_ids),:);
% 绘制 x 坐标随时间变化图
figure;
subplot(2,1,1);
for i=1:length(vehicle_ids)
    idx=selected_data.vehicle_id==vehicle_ids(i);
    plot(selected_data.time(idx),selected_data.x(idx));
    hold on;
end
xlabel('Time');
ylabel('X-coordinate');
legend('Vehicle 18','Vehicle 32','Vehicle 54');
% 绘制 y 坐标随时间变化图
subplot(2,1,2);
for i=1:length(vehicle_ids)
    idx=selected_data.vehicle_id==vehicle_ids(i);
    plot(selected_data.time(idx),selected_data.y(idx));
    hold on;
```

```

end
xlabel('Time');
ylabel('Y-coordinate');
legend('Vehicle 18','Vehicle 32','Vehicle 54');
%%绘制 A1 路口车辆编号为 18, 32, 54 的速度变化图
data1=readtable('D:\新建文件夹\2024 华中杯赛题\B 题赛题\附件\附件 1\A1.csv');
% 提取车辆 18、32 和 54 的数据
vehicle18=data1(data1.vehicle_id==18,:);
vehicle32=data1(data1.vehicle_id==32,:);
vehicle54=data1(data1.vehicle_id==54,:);
% 计算速度
vehicle18_speed=sqrt(diff(vehicle18.x).^2+diff(vehicle18.y).^2)./diff(vehicle18.time);
vehicle32_speed=sqrt(diff(vehicle32.x).^2+diff(vehicle32.y).^2)./diff(vehicle32.time);
vehicle54_speed=sqrt(diff(vehicle54.x).^2+diff(vehicle54.y).^2)./diff(vehicle54.time);
% 画出速度曲线
hold on;
subplot(3,1,1);
plot(vehicle18.time(2:end),vehicle18_speed,'r','LineWidth',1);
legend('车辆 18')
subplot(3,1,2);
plot(vehicle32.time(2:end),vehicle32_speed,'g','LineWidth',1);
legend('车辆 32')
subplot(3,1,3);
plot(vehicle54.time(2:end),vehicle54_speed,'b','LineWidth',1);
legend('车辆 54')
xlabel('时间');
ylabel('速度');
title('车辆速度曲线');

Q1-2:计算车辆红灯时间和周期模型

%%
clc,clear
% 数据读取

```

```

data1=readtable('A1.csv');
%预处理，检查填补缺失值
data1=fillmissing(data1,'linear','DataVariables',{ 'x' , 'y' });
vmin = 0; % 速度阈值
vmax = 5;
clear
%%加载数据
%%绘制 A1 路口所有车辆轨迹
data1=readtable('A1.csv');
gscatter(data1.x, data1.y, data1.vehicle_id,'bryckm','.', 10);
xlabel('x');
ylabel('y');
title('C1 路口车辆行驶轨迹');
% 计算车辆速度和状态
% 对每个车辆进行循环处理
unique_vehicles = unique(data1.vehicle_id); % 获取所有独特的车辆 ID
for i = 1:length(unique_vehicles)
    % 提取当前车辆的数据
    current_vehicle = unique_vehicles(i);
    vehicle_data = data1(data1.vehicle_id == current_vehicle, :);
    % 对时间进行排序（确保数据按时间顺序）
    vehicle_data = sortrows(vehicle_data, 'time');
    % 提取 x, y 坐标和时间
    t = vehicle_data.time;
    x = vehicle_data.x;
    y = vehicle_data.y;
    % 计算空间距离差
    dx=diff(x);
    dy=diff(y);
    dt=diff(t);
    % 计算速度
    v = sqrt(dx.^2 + dy.^2) ./ dt;
    % 绘制速度时间图
    plot(t(2:end), v, 'DisplayName', sprintf('Vehicle %d', current_vehicle));
    hold on; % 保持图像，对下一个车辆绘制在同一图上
end
% 设置图例和标签
legend show;
xlabel('Time');
ylabel('Velocity (m/s)');
title('车辆速度随时间变化曲线');
hold off;
%%计算所有车辆的红灯时间
unique_vehicles=unique(data1.vehicle_id);

```

```

red_lights=[]
for i = 1:length(unique_vehicles)
    % 提取当前车辆的数据
    current_vehicle = unique_vehicles(i);
    vehicle_data = data1(data1.vehicle_id == current_vehicle, :);
    % 对时间进行排序（确保数据按时间顺序）
    vehicle_data = sortrows(vehicle_data, 'time');
    % 提取 x, y 坐标和时间
    t = vehicle_data.time;
    x = vehicle_data.x;
    y = vehicle_data.y;
    % 计算空间距离差
    dx=diff(x);
    dy=diff(y);
    dt=diff(t);
    % 计算速度
    v = sqrt(dx.^2 + dy.^2) ./ dt;
    % 寻找绿灯和红灯的时刻
    v0=find(v==0);%寻找车速为 0
    v5=find(v<=5)%寻找车速等于 5
    if ~isempty(v0) && ~isempty(v5)
        % 红灯结束时刻：速度从 0 开始增加
        red_t_end=t(v0(end));
        % 红灯时刻：速度减到 5 的时刻中的最大值
        red_t_start=t(v5(1));
        % 计算红灯时间
        if v5(1)<v0(end);
            red_light=(red_t_end-red_t_start);
            %red_lights=[];% 用于存储所有车辆的红灯时间
            if red_light>2;
                red_lights = [red_lights, red_light];
            end
        end
    end
end
mean_red_light=mean(red_lights)

```

Q1-3:计算车辆红灯时间和周期优化模型

```

clear
close all
clc
A1 = xlsread("A5.csv");
%a = readtable("A1.csv");
%hashtable = containers.map();

```

```

AllCar = A1(:,2);
CarNumber = [];
for i=1:size(AllCar)          %找出表内所有 car 的 id
    if ~ismember(AllCar(i),CarNumber)
        CarNumber = [CarNumber,AllCar(i)];
    end
end
%% for i= 1:size(CarNumber')  %通过 car ID 提取对应的 time ID x y 信息
%%     name = ['car',num2str(CarNumber(i))];
%%     temp = [];
%%     for j = 1:size(AllCar)
%%         if CarNumber(i)==A1(j,2)
%%             temp = [temp;A1(j,:)];
%%         end
%%     end
%%     eval([name ' =temp;']);
%%     clear name
%%     clear temp
%% end
Time = [];
%plot(0);
starTime =[];
brkTime = [];
%hold on
for i= 1:size(CarNumber')    %ID 对每辆车提取时间和 xy 位移
    name = ['car',num2str(CarNumber(i))];
    temp = [];
    for j = 1:size(AllCar)
        if CarNumber(i)==A1(j,2)
            temp = [temp;A1(j,:)];
        end
    end
    speedx = diff(temp(:,3));
    speedy = diff(temp(:,4));
    deta = [speedx,speedy];
    [time_of_x,starx,breakx] = searchbeginToEnd(speedx);
    [time_of_y,stary,breaky] = searchbeginToEnd(speedy);
    if starx <= 0
        tex = -1;
    else
        tex = temp(starx,1);
    end
    if stary <= 0

```

```

        tey = -1;
    else
        tey = temp(stary,1);
    end
    if breakx >= 0
        brx = 0;
    else
        brx = temp(starx-breakx-time_of_x,1);
    end
    if breaky >= 0
        bry = 0;
    else
        bry = temp(stary-breaky-time_of_y,1);
    end
    brkXY = [brx;bry];
    starXY = [tex;tey];
    starTime = [starTime,starXY];
    brkTime = [brkTime,brkXY];
    temp_of_time =[CarNumber(i);time_of_x;time_of_y];
    Time = [Time,temp_of_time];
    %%A1 只有 X 的位移，所以可以只判断 X 轴
    %%     tempp = [CarNumber(i);time];
    %%     Time = [Time,tempp];
    deta = [deta;time_of_x,time_of_x];
    temp = [temp,deta];
    eval([name ' =temp;']);
    clear name
    clear temp
end
TimeX_desc = sort(Time(2,:), 'descend');
TimeY_desc = sort(Time(3,:), 'descend');
TimeX_desc(1:5);
TimeY_desc(1:5);
Time_above_0 = [];
for i= 1:size(Time')
    if Time(2,i)>0
        temp = Time(:,i);
        Time_above_0 = [Time_above_0,temp];
    end
end
end
%从图里可以看出，其实用起步时间才是比较合理的，因为红灯时，各个车离路口远近不一样，
%所以刹车时间点不同
%scatter(brkTime(1,:),1.1, '*');
hold on

```

```

scatter(startTime(1,:),1,'*');
%handle_starTime = sort(startTime(1:),'descend');
handle_starTime = startTime(1:);
%%对于 startTime 进行处理，每个周期内找出第一台起步车，并且以这辆车为基准
for i = 2: size(handle_starTime')
    if abs(handle_starTime(1,i) - handle_starTime(1,i-1)) < 20
        handle_starTime(1,i) = handle_starTime(1,i-1);
    end
end
scatter(handle_starTime(1,:),1,'*');
title('A5 路口车辆起步时刻');
f_of_starTime = diff(handle_starTime(1,:));
f_of_starTime = f_of_starTime(f_of_starTime~=0);
f_of_starTime = f_of_starTime(f_of_starTime~-1);
zhouqi = mode(f_of_starTime);
s1 = ['周期为',num2str(zhouqi),'秒'];
disp(s1);
%%根据 handle_starTime 的起始时刻重新修正每辆车灯红灯的时长
xiuzhen = handle_starTime - brkTime;
xiuzhen_MAX = sort(xiuzhen(1:),'descend');
xiuzhen_MAX(1:5)

%% 子函数 1

function [outputArg1,last_0,under_throuhlod] = searchbeginToEnd(speed)
%SEARCBEGINTOEND
%输入速度的向量，找到连续的 0，并且向前搜索第一个大于阈值的速度，并且向后搜索最后一个 0
%返回这样的个数，并认为这个就是时间。last_0 返回的是起步时刻,返回的只是相对时间。
thrhold = 2; %阈值速度
last_0 = -1; %如果没有起步时刻 返回-1
under_throuhlod = 1;
size_of_speed = size(speed);
numbe_of_0 = sum (speed==0);
if(numbe_of_0 < 10) %等红灯时间不到 10 秒。直接返回等红灯 0 秒，他不是头车
    outputArg1 = 0;
    return
end
if(size_of_speed-numbe_of_0 < 10) %有些车一直沿一个方向移动，有些方向几乎没有
    outputArg1 = -1; %速度，就会造成 0 的个数和数据长度差不多，此时返回-1
    return
end
location_of_0 = find(speed==0);
time = location_of_0(end) - location_of_0(1);
last_0 = location_of_0(end);

```

```

if(time >= 80)      %如果等红灯时间超过 80 秒，这个数据就有问题了，抛-2 待查
    outputArg1 = -2;
    return
end
first_0 = location_of_0(1);
if first_0<10      %如果第一个 0 开始的时间很早 说明在记录这条数据之前已经在等红灯了，不完整
    outputArg1 = 0;
    return
end
time_of_above = 0;
for i = 1:10      %只允许从第一个 0 开始往前搜 10 秒
    if abs(speed(first_0-i)) < thrhold
        time_of_above = i;
    end
end
time = time +time_of_above;
under_throuhloclod = -time_of_above;
outputArg1 = time;
end

```

Q1-4:判定红灯时长说明

```

clear
%%判定红灯时长图片
data1=readtable('A1.csv');
% 提取编号为 18 的车辆的数据
vehicle18=data1(data1.vehicle_id==18,:);
% 计算速度
distance=sqrt(diff(vehicle18.x).^2+diff(vehicle18.y).^2); % 计算相邻位置之间的距离
time_diff=diff(vehicle18.time); % 计算相邻时间之间的差值
speed=distance./time_diff; % 计算速度
% 画出速度随时间变化的曲线
plot(vehicle18.time(2:end),speed);
xlabel('时间');
ylabel('速度');
title('18 车速度随时间变化');
hold on
x=60:0.1:95;
y=5*ones(size(x));

```



```

plot(x,y,'DisplayName','y=5');
legend
%%绘制 A1 路口等红灯时间箱线图
data1=readtable('D:\新建文件夹\附录\A1 路口车辆等红灯时间.xlsx');
boxplot(data1.t);
ylabel('时间(s)');
title('等红灯时间(优化前)');
figure
data2=readtable('D:\新建文件夹\附录\A1 路口车辆等红灯时间（优化模型）.xlsx');
boxplot(data2.t);
ylabel('时间(s)');
title('等红灯时间(优化后)');

```

第二问：讨论某些因素对模型估计精度的影响，求 B 路口的信号灯周期

依据第一问的优化模型即可求解 B 路口的红绿灯时长

依据第一问的优化模型结合以下筛选出的数据，可探讨车辆比例，车流量以及定位误差对于模型估计精度的影响。

Q2-1:车辆比例的影响

```

%%提取 B1 中特定 18 辆车的数据
%%减少车辆比例
data1=readtable('B1.csv');
% 假设你想要保留的特定车辆编号为 30, 18, 56, 80
vehicle_ids=[30,107,259,329,405,425,456,521,583,625,755,823,987,1114,1199,1457,1491,1566];
% 根据特定车辆编号筛选数据
logical_index=ismember(data1.vehicle_id,vehicle_ids);
vehicle_data=data1(logical_index,:);
% 存储到名为 B1-18 辆车的数据的表格中
writetable(vehicle_data, 'B1-18 辆车的.xlsx');
%%

```

Q2-2:车流量的影响

```
%%提取前二十分钟，20-40 分钟，后二十分钟的数据
%%减少车流量

data1=readtable('B1.csv');
% 提取时间为 0s 到 1200s 的数据
data_20=data1(data1.time>=0&data1.time<=1200,:);
% 存储到名为 B1-前二十分钟的表格中
writetable(data_20,'B1-前二十分钟.xlsx');
% 提取时间为 1247s 到 2423s 的数据
data_20_40=data1(data1.time>=1200&data1.time<=2400,:);
% 存储到名为 B1-20-40 分钟的表格中
writetable(data_20_40,'B1-20-40 分钟.xlsx');
% 提取时间为 2400s 到 3599s 的数据
data_40_60=data1(data1.time>=2400&data1.time<=3599,:);
% 存储到名为 B1-40-60 分钟的表格中
writetable(data_40_60,'B1-40-60 分钟.xlsx');
```

Q2-3:定位误差的影响

```
%%%%%计算定位误差的影响
% 导入 data1 表格
data1 = readtable('B1.csv');
% 第一组 x,y 坐标变化车辆编号
increase_vehicles = [30, 44,
56,106,107,142,168,194,259,268,314,325,329,344,380,402,405,414,420,424,425,
436,440,441,456,480,485,496,521,545,561,581,583,586,594,620];
% 第二组 x,y 坐标变化车辆编号
decrease_vehicles =
[625,628,644,666,668,708,715,743,745,748,755,757,765,766,823,913,938,987,99
5,1002,1094,1102,1114,1117,1123,1178,1199,1340,1425,1434,1457,1459,1465,148
1,1491,1544,1566];
% 复制 data1 表格到 data2 表格
data2 = data1;
% 循环遍历需要变化的车辆编号
for i=1:length(increase_vehicles)
```

```

% 获取第一组的车辆编号
vehicle_id = increase_vehicles(i)
% 获取该车辆的 x 和 y 坐标
x=data2.x(data2.vehicle_id==vehicle_id);
y=data2.y(data2.vehicle_id==vehicle_id)
x_new=x+5%*x;
y_new=y+5%*x;
% 将增加后的坐标更新到 data2 表格中
data2.x(data2.vehicle_id==vehicle_id)=x_new;
data2.y(data2.vehicle_id==vehicle_id)=y_new;
end
for i=1:length(decrease_vehicles)
% 获取第二组的车辆编号
vehicle_id=decrease_vehicles(i);
% 获取该车辆的 x 和 y 坐标
x=data2.x(data2.vehicle_id==vehicle_id);
y=data2.y(data2.vehicle_id==vehicle_id);
% 减少 5% 的 x 和 y 坐标
x_new=x-5%*x;
y_new=y-5%*X;
% 将减少后的坐标更新到 data2 表格中
data2.x(data2.vehicle_id==vehicle_id)=x_new;
data2.y(data2.vehicle_id==vehicle_id)=y_new;
end
% 将 data2 表格保存到 csv 文件中
writetable(data2, 'data2.csv');
data1=readtable('B1.csv');
% 假设你想要保留的特定车辆编号为 30, 18, 56, 80
vehicle_ids=[30,107,259,329,405,425,456,521,583,625,755,823,987,1114,1199,1
457,1491,1566];
% 根据特定车辆编号筛选数据
logical_index=ismember(data1.vehicle_id,vehicle_ids);
vehicle_data=data1(logical_index,:);
% 存储到名为 B1-18 辆车的数据的表格中
writetable(vehicle_data, 'B1-18 辆车的数.据.xlsx');

```

第三问：判断 C 路口信号灯周期是否变化，求周期切换的时刻以及新旧周期参数

Q3-1:动态检测周期

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import find_peaks

def load_and_preprocess_data(filepath):
    data = pd.read_csv(filepath)
    data['speed'] = np.sqrt(data.groupby('vehicle_id')[['x', 'y']].diff().p
ow(2).sum(axis=1))
    data['stopped'] = data['speed'] == 0
    data['stop_change'] = data.groupby('vehicle_id')['stopped'].diff()
    return data

def calculate_stop_durations(data):
    stop_times = data[data['stop_change'] == 1]
    start_times = data[data['stop_change'] == -1]
    stop_times['next_start_time'] = stop_times.groupby('vehicle_id')['time'
].shift(-1)
    stop_times['duration'] = stop_times['next_start_time'] - stop_times['ti
me']
    return stop_times['duration'].dropna().values

def plot_histogram(data, title, xlabel, ylabel, bins, color, edgecolor):
    plt.figure(figsize=(10, 5))
    plt.hist(data, bins=bins, color=color, edgecolor=edgecolor)
    plt.title(title)
    plt.xlabel(xlabel)
    plt.ylabel(ylabel)
    plt.grid(True)
    plt.show()

def cusum(data, threshold=1.5, drift=0):
    mean_data = np.mean(data)
    S, S_max, change_points = 0, 0, []
    for i in range(1, len(data)):
        increment = data[i] - mean_data - drift
        S = max(0, S + increment)
        if S > S_max + threshold:
            change_points.append(i)
            S_max = S
    return change_points
```

```

data = load_and_preprocess_data('C1.csv')
stop_durations = calculate_stop_durations(data)
plot_histogram(stop_durations, 'Histogram of Stop Durations', 'Duration (seconds)', 'Frequency', 30, 'yellow', 'black')
average_duration = np.mean(stop_durations)
peaks,=find_peaks(stop_durations, height=average_duration)
peak_durations = stop_durations[peaks]
plt.figure(figsize=(10, 5))
plt.plot(stop_durations, 'o', alpha=0.5, label='Stop Durations')
plt.plot(peaks, peak_durations, 'ro', label='Peaks above Average')
plt.title('Peaks in Stop Durations')
plt.xlabel('Index')
plt.ylabel('Duration (seconds)')
plt.legend()
plt.grid(True)
plt.show()
change_points = cusum(peak_durations, threshold=10, drift=0)
print("变化点索引: ", change_points)

```

第四问：综合轨迹数据分析，使用附件 4 中某路口连续 2 小时内所有方向样本车辆的轨迹数据来识别出该路口信号灯的周期。

Q4-1 车辆行驶方向分类

```

clear
close all
clc

A1 = xlsread("D.csv");

%load('A1.mat');
R_hold = 420;
% R_hold = 0;
%
% center_x = 0;
% center_y = 0;
% radius = 420;
%
% % 画圆
% figure; % 创建新图形窗口
% rectangle('Position',[center_x - radius, center_y - radius, radius * 2, radius * 2], ...

```

```

%      'Curvature',[1,1], 'EdgeColor','r', 'LineWidth',2);

AllCar = A1(:,2);
CarNumber = [];
for i=1:size(AllCar)      %找出表内所有 car 的 id
    if ~ismember(AllCar(i),CarNumber)
        CarNumber = [CarNumber,AllCar(i)];
    end
end

hold on

%%这里的 12 种情况只对应了每个路口的三个行进方向。
case1 = [];
case2 = [];
case3 = [];
case4 = [];
case5 = [];
case6 = [];
case7 = [];
case8 = [];
case9 = [];
case10 = [];
case11 = [];
case12 = [];

staLocation = [];
endLocation = [];

for i= 1:size(CarNumber')      %ID 对每辆车提取时间和 xy 位移
    name = ['car',num2str(CarNumber(i))];
    temp = [];
    for j = 1:size(AllCar)
        if CarNumber(i)==A1(j,2)
            temp = [temp;A1(j,:)];
        end
    end
    if size(temp)<=60      %记录少于 60 条的删去
        continue;
    end
    R = (temp(end,3)^2+temp(end,4)^2).^0.5;
    if R < R_hold      %endlocation 在小圆内的也删去
        continue;
    end
end

```

```

end

temp_star = [temp(1,2);temp(1,3);temp(1,4)];
staLocation = [staLocation,temp_star];
temp_end = [temp(end,2);temp(end,3);temp(end,4)];
endLocation = [endLocation,temp_end];
clear temp_star
clear temp_end
% plot(temp(1,3),temp(1,4),'*');
% plot(temp(end,3),temp(end,4),'*')
% plot(temp(1,3),temp(1,4),'*');
end
plot(staLocation(2,:),staLocation(3:),'r*')
%plot(endLocation(2,:),endLocation(3:),'*')
out = staLocation-endLocation;
out = out';

for i = 1:size(staLocation')
    %case123
    if (staLocation(2,i)==495.39) && (endLocation(2,i) <=-400)
        case2 = [case2,staLocation(1,i)];
    end
    if (staLocation(2,i)==495.39) && (endLocation(3,i) > 400)
        case1 = [case1,staLocation(1,i)];
    end
    if (staLocation(2,i)==495.39) && (endLocation(3,i) <-400)
        case3 = [case3,staLocation(1,i)];
    end
    %case456
    if (staLocation(2,i)==48.27) && (endLocation(2,i) <=-400)
        case4 = [case4,staLocation(1,i)];
    end
    if (staLocation(2,i)==48.27) && (endLocation(3,i) < -400)
        case5 = [case5,staLocation(1,i)];
    end
    if (staLocation(2,i)==48.27) && (endLocation(2,i) > 400)
        case6 = [case6,staLocation(1,i)];
    end
    %case789
    if (staLocation(2,i)==-496.22) && (endLocation(3,i) >= 400)
        case7 = [case7,staLocation(1,i)];
    end
    if (staLocation(2,i)==-496.22) && (endLocation(2,i) > 400)

```

```

        case8 = [case8,staLocation(1,i)];
    end
    if (staLocation(2,i)==-496.22) && (endLocation(3,i) < -400)
        case9 = [case9,staLocation(1,i)];
    end
    %case 10 11 12
    if (staLocation(2,i)==-48.27) && (endLocation(2,i) < -400)
        case10 = [case10,staLocation(1,i)];
    end
    if (staLocation(2,i)==-48.27) && (endLocation(3,i) > 400)
        case11 = [case11,staLocation(1,i)];
    end
    if (staLocation(2,i)==-48.27) && (endLocation(2,i) > 400)
        case12 = [case12,staLocation(1,i)];
    end
end
end

```

Q4-2 计算信号灯周期

```

%%计算信号灯周期
A1 = xlsread('D.csv');
clear
close all
clc;
%load('D.mat');
T = case1 ;%%需要手工把 case1-case12 放到 T 里面。每一次都会返回
    %该方向的周期、以及红灯时间。

case2_cell = [];
Time = [];
starTime = [];
brkTime = [];
case2 = T';
%case3 = case3';
for i = 1:size(case2)
    temp_of_case2 = find(A1(:,2)==case2(i));
    name = ['car',num2str(case2(i))];
    temp = [];
    for j = 1:size(temp_of_case2)
        temp = [temp;A1(temp_of_case2(j),:)];
    end
    speedx = diff(temp(:,3));
    speedy = diff(temp(:,4));
    deta = (speedx.^2+speedy.^2).^0.5;

```



```

[time_of_x,starx,breakx] = searchbeginToEnd(deta);
if starx <= 0
    tex = -1;
else
    tex = temp(starx,1);
end
if breakx >= 0
    brx = 0;
else
    brx = temp(starx-breakx-time_of_x,1);
end
brkXY = [brx];
starXY = [tex];
starTime = [starTime,starXY];
brkTime = [brkTime,brkXY];
temp_of_time =[case2(i);time_of_x];
Time = [Time,temp_of_time];
deta = [deta;time_of_x];
temp = [temp,deta];

eval([name ' =temp;']);
end

%plot(starTime,1,'*');

TimeX_desc = sort(Time(2:),'descend');
%TimeY_desc = sort(Time(3:),'descend');
TimeX_desc(1:5)
%TimeY_desc(1:5)
Time_above_0 =[];
for i= 1:size(Time')
    if Time(2,i)>0
        temp = Time(:,i);
        Time_above_0 = [Time_above_0,temp];
    end
end

handle_starTime = starTime(1,:);
%%对于 starTime 进行处理，每个周期内找出第一台起步车，并且以这辆车为基准
for i = 1: size(starTime')-1    %维护左指针和右指针，指针范围内的数，只要差小于
20,
                                %就都将起始时刻改为

    left = i;
    right = i+1;

```

```

    if handle_starTime(1,left) - handle_starTime(1,right) <= 22
        right = right + 1;
    else
        handle_starTime(1,left:right-1) = handle_starTime(1,left);
        i = right - 1;
        continue
    end
end
end
scatter(handle_starTime(1,:),1,'*');
hand_of_F_time=sort(handle_starTime,'descend');
f_of_starTime = diff(hand_of_F_time(1,:));
f_of_starTime = f_of_starTime(f_of_starTime~=0);
f_of_starTime = -f_of_starTime(f_of_starTime~-1);
f_of_starTime = f_of_starTime(f_of_starTime>=10);
zhouqi = mode(f_of_starTime);
s1 = ['周期为',num2str(zhouqi),'秒'];

disp(s1);

xiuzhen = handle_starTime - brkTime;
xiuzhen_MAX = sort(xiuzhen(1:5),'descend');
xiuzhen_MAX(1:5)

%%这个版本是特殊的 searchbeginToEnd。红灯等待时长被容许到了 120 秒而不是 80 秒。
%%以此解决返回-2 的问题。
function [outputArg1,last_0,under_throuhlod] = searchbeginToEnd(speed)
%SEARCBEGINTOEND
%输入速度的向量，找到连续的 0，并且向前搜索第一个大于阈值的速度，并且向后搜索最后一个 0
%返回这样的个数，并认为这个就是时间。last_0 返回的是起步时刻,返回的只是相对时间。

thrhold = 11;    %阈值速度
last_0 = -1;    %如果没有起步时刻 返回-1
under_throuhlod = 1;

size_of_speed = size(speed);
numbe_of_0 = sum (speed==0);
if(numbe_of_0 < 10)    %等红灯时间不到 10 秒。直接返回等红灯 0 秒，他不是头车
    outputArg1 = 0;
    return
end

if(size_of_speed-numbe_of_0 < 10)    %有些车一直沿一个方向移动，有些方向几乎没有
    outputArg1 = -1;    %速度，就会造成 0 的个数和数据长度差不多，此时返回-1

```

```

        return
    end

    location_of_0 = find(speed==0);
    time = location_of_0(end) - location_of_0(1);
    last_0 = location_of_0(end);
    if(time >= 120)          %如果等红灯时间超过 80 秒，这个数据就有问题了，抛-2 待查
        outputArg1 = -2;
        return
    end
    first_0 = location_of_0(1);

    if first_0<10          %如果第一个 0 开始的时间很早 说明在记录这条数据之前已经在等红
        灯了，不完整
        outputArg1 = 0;
        return
    end

    time_of_above = 0;
    for i = 1:10          %只允许从第一个 0 开始往前搜 10 秒
        if abs(speed(first_0-i)) < thrhold
            time_of_above = i;
        end
    end
    time = time +time_of_above;
    under_throuhlod = -time_of_above;

    outputArg1 = time;

end

```