基于绿色区间估计的交通信号灯估计模型

摘要

在城市路网管理和智能驾驶系统的开发中，准确获取交通信号灯的周期参数至关重要。在无直接接入的信号灯数据时，利用客户的行车轨迹数据估计交通信号灯的参数成为了一项主要的研究方法。针对这一问题，本文基于车辆轨迹信息，建立了一种有效估计路口交通信号灯红绿周期的方法。该方法主要包括本文所提出的两个模型：**GCD-KS 模型和绿色区间估计模型**。前者用于估计交通信号灯的完整周期，后者用于估计一个完整周期内的绿灯时间。通过此方法，本文在四类数据集上均取得了不错的估计效果，对城市路网中交通信号灯的红绿周期的获取有重要价值。

针对问题 1，本文先对数据进行了可视化处理，发现其中记录的车辆的行驶轨迹有直线和转弯两种模式，且单个数据集且表现出一种模式。根据车辆坐标展现出的规律，发现车辆在某一路口中只有一个停车点，即为路口位置。基于此定义并统计了通过事件和停车事件，然后应用 **GCD-KS 模型和绿色区间估计模型**，得出了各个路口的红绿灯周期。**如路口 A1 的红灯时长为 75s，绿灯时长为 30s**。

针对问题 2，通过分析数据集发现，虽然每个数据集记录的车辆行驶轨迹模式不变，但是相较于问题 1 的数据集，车辆数量明显减少，停车点位置增多。因此本文采用 DBSCAN 密度聚类算法，提取密度最高的停车点作为路口位置，接着分析了所有通过事件和停车事件，应用 **GCD-KS 模型和绿色区间估计模型**，得出了各个路口的红绿灯周期，如**路口 B4 的红灯时长为 80s，绿灯时长为 25s**。同时本文还讨论了不同因素对上述模型估计结果精度的影响，得出**采样比例和车流量过小容易导致绿色区间估计不显著，进而导致估计的绿灯时间偏小，而定位误差可能会造成异常点**。

针对问题 3，针对红绿灯周期可能存在的变化，本文分别采用了**移动窗口 GCD-KS 模型**和**差分方法**进行检测，两种方法都表明信号灯整体周期不变，进而说明周期的变化可能是红绿灯时间的占比。然后本文提取了数据集中所有的停车时长序列，利用**移动平均差分**的方法，检测其绝对值是否存在峰值以识别潜在的变点。针对存在的变点，在其两侧分别运用绿色区间估计得出红绿灯时长。**结果显示，只有部分路口的红绿灯时间发生了变化，比如路口 C2 的绿灯时间在 3567s 时从 15s 下降到了 10s**。

针对问题 4，本文通过绘制散点图，观察到该路口各个方向都有车辆的行驶轨迹。因此我们使用 **K-means 聚类算法**，将车辆轨迹划分为 12 类，然后对每一种行驶模式采用 **GCD-KS 模型和绿色区间估计模型**，得出该路口**各个信号灯的红绿周期均为 142s**。再基于十字路口的信号灯之间的关联，归纳出了信号灯的四个变换阶段。进一步地，利用**全向联合周期确定法**确定了**路口信号灯四个阶段的切换时间分别是 49s，24s，27s 和 42s**。

关键词：交通信号灯预测；绿色区间估计；K-S 检验；最大公约数求解；时间序列变点检测；聚类算法。

目录

一　问题背景与重述

1.1　问题背景

获取城市路网中的交通灯的红绿周期对于电子地图服务商、智能驾驶方案提供商等至关重要。尽管有部分信号灯的各项参数已经接入了网络，可以从网络中直接获取，但是出于成本问题或安全问题，仍有大量信号灯未接入网络。人工读取路口信号灯的各项参数不仅成本高昂，也不可能覆盖所有信号灯。因此，在无法直接获取信号灯参数的情况下，利用客户的行车轨迹数据估计交通信号灯的参数成为了一项主要的研究方法。像这种，利用 GPS 定位功能采集车辆的位置和时间信息进而去估计交通流发方法，成为浮动车辆数据技术（FCD）.

基于浮动车辆数据去预测信号灯的周期，是浮动车辆数据的一大应用场景。通过准确预测信号灯的变化情况，电子地图服务商可以改进线路图，提供实时交通信息，以及预测可能的交通堵塞，

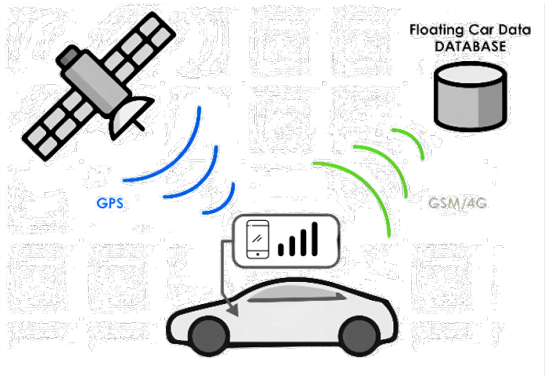


图 1.1: 浮动车辆数据原理

1.2　问题重述

问题的核心是利用车辆轨迹参数去估计路口交通信号灯的周期，各问题的差异由不同特征的数据集体现。

- 问题 1：在信号灯**周期固定不变**且已知**所有**车辆的行车轨迹的条件下估计路口信号灯的红绿周期.
- 问题 2：在信号灯周期固定不变且只有**部分**样本车辆的行车轨迹条件下估计路口信号灯的红绿周期，并分析车辆比例、车流量、定位误差对估计模型的影响.
- 问题 3：在**信号灯红绿周期可能发生变化**的条件下识别出周期切换时刻以及新旧红绿周期的参数.
- 问题 4：给出某一路口**所有方向**样本车辆的轨迹数据，估计路口信号灯的红绿周期

二　问题分析

考虑到所有的问题都涉及到通过车辆轨迹数据去估计红绿灯周期，我们参考了 Markus Kerper,Yi-Ta Chuang 等人的的相关工作[1][2][3]，结合了题目所给数据集的特征，建立了 **GCD-KS 模型**和**绿色区间估计模型**。

2.1　问题 1 的分析

问题 1 要求根据附件 1 给出的 A1-A5 五个路口的车辆轨迹数据，估计五个路口各自的红绿灯周期。本文先绘制出了 A1-A5 五个路口所有车辆坐标的散点图，发现 A1,A2,A5 路口记录的是直行方向车辆的轨迹，而 A3,A4 路口记录的是转弯方向的车辆。通过数据分析发现了五个路口车辆的停车位置都较为集中，由此推断停车位置即为路口位置。记录车辆通过路口的时刻和停车时刻，先利用 **GCD-KS 模型**得出五个路口的红灯——绿灯周期，再利用**绿色区间估计模型**得出绿灯时间和红灯时间。

2.2　问题 2 的分析

问题 2 与问题 1 的数据集高度相似，但附件 2 给出的 B1-B5 路口的车辆轨迹数据存在车辆停车点较多、车辆数据少等问题。我们先通过 **DBSCAN 密度聚类算法**，以最密的停车点的中心点作为路口位置，接着利用 **GCD-KS 模型**得出五个路口的红灯——绿灯周期。对于远离路口的停车事件，根据离路口的远近分别赋予不同的权重考虑，利用改进的**绿色区间估计模型**得出绿灯时间和红灯时间。

2.3　问题 3 的分析

题目告知了问题 3 的路口红绿周期可能存在变化。我们先通过停车点找到路口位置，记录汽车在路口第一次启动的时刻。通过差分发现这个值是个定值，因此我们合理推测完整周期并未改变，改变的是红绿灯时间的占比，接着利用**移动窗口的 GCD-KS 模型**验证了我们的想法。为了寻找周期改变点，我们对停车时间序列进行了移动平均，再进行一阶差分，通过峰值寻找数据的变点。最后，我们利用**绿色区间估计模型**得出了变点两侧的红绿周期。

2.4　问题 4 的分析

问题 4 需要我们分析在同一个路口不同方向的轨迹数据，我们使用 K-means 算法对所有数据进行聚类，得到 12 类车辆行驶轨迹，因此可以将此问题简化为前面问题的模型，使用 GCD-KS 模型和绿色区间模型估计计算得到所有方向的周期和红绿周期。考虑到同一路口各个方向的信号灯周期都是相互关联的，我们将使用**全向联合周期确定法 (all- direction joint determination method)**[4]得到该路口的信号灯周期。

三　问题假设与符号说明

3.1　问题假设

针对上述问题，为了建立模型，我们提出如下的假设：

1. 附件所有的信息都是真实可靠的，即轨迹信息都是真实的车辆行为。

   针对假设 1，由于给出的数据十分稀疏 (A 类数据每小时平均仅 96 辆车)，再剔除数据会使得分析变得困难，所以我们必须尽量保留数据，前提是认为数据都是可靠的。

2. 数据记录的车辆里不存在大量的异常行为：如闯红灯行为、路口靠边停车等。

   针对假设 2，是为了保证红绿灯的估计的精度，如果有太多的闯红灯和异常停车，则精度无法保证。

3. 对记录车辆的选择无偏好性；选择的路口也无偏好性。

   针对假设 3，是保证红绿灯周期的时长以及切换符合一般性规律，以舍弃一些不符合常识的，怪异的解。

3.2　名词说明

- **周期：**在文中指的是一个完整的周期，等于一个红灯时间和一个绿灯时间之和.
- **红灯周期（红灯时间）**：一个红灯的时间.
- **绿灯周期（绿灯时间）：**一个绿灯的时间.
- **停车事件：**车辆在某时刻发生了停车
- **通过事件：**车辆在某时刻通过了路口红绿灯

3.3　符号说明

符号	说明
v_i	第 i 辆车
x_{im}	第 i 辆车在时刻 m 的横坐标
y_{im}	第 i 辆车在时刻 m 的纵坐标
t_i^s	第 i 辆车停下的时刻
t_i^c	第 i 辆车穿越路口的时刻
t_i^{start}	第 i 辆车等待完红灯启动的时刻
C	一个完整周期，即红灯与绿灯时间之和
C_G	一个完整周期内的绿灯时间
C_R	一个完整周期内的绿灯时间

四　两个基本模型的建立

在考虑到两个模型在所有问题求解中的普遍应用性，本文将它们单独划分为一个独立的章节进行详细论述。这样的处理不仅有助于突出这两个模型的核心地位和理论价值，而且也为后续各个具体应用场景中的问题分析和求解提供了统一的理论基础和分析工具。

4.1　基于模型的数据预处理

考虑到两个模型需要正确识别三类事件：

- 停车事件：车辆发生停车的时刻
- 通过事件：车辆通过路口的时刻
- 启动事件：车辆在路口停下后，再次启动的时刻

其中，对于停车事件，只需要检测车辆坐标的变化即可，如果在 t 时刻和 $t+1$ 时刻，车辆的 x, y 坐标变化均小于 0.1，我们就认为车辆在 t 时刻发生了停车事件。

对于通过事件，首先要能够正确的识别路口位置。路口位置一般就是停车最密集点的，对于停车点较少的，可以简单识别；停车点较多的，可以采用密度聚类方法。其次要能够识别出车辆跨越路口的时刻，考虑到变道现象或者双车道，我们需要用 x 和 y 两个坐标确定车辆是否通过了路口，因此对于每个车道，需要编写不同的判断条件，而不能统一套用。

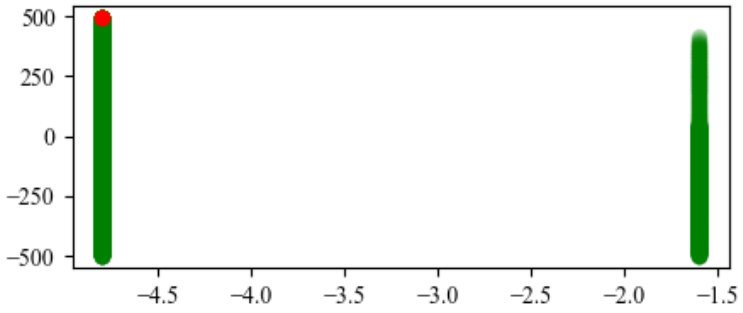


图 4.2: 典型的双车道：来自 C3

对于启动事件，一旦正确判断了停车事件，只需要寻求路口附近的停车事件，设定一个阈值，将离路口距离小于阈值的停车事件都视为路口停车，那么其下一次坐标变化的时刻就是启动事件。

4.2　求解完整周期的 GCD-KS 模型

本题要求解的是对应信号灯的红灯周期和绿灯周期，为了方便求解，我们首先需要求解出一个完整周期 C，之后再求出绿灯周期 C_G（或者红灯周期 C_R），就可以得到另一种颜色信号灯的周期。为此，我们建立了 **GCD-KS 模型**，模型的误差可以控制在 1s 以内。

4.2.1 模型的前置条件验证

模型需要满足一个基本条件，即车辆的到达在时间上要满足**均匀分布**，针对这个问题，我们可以利用 Kolmogorov-Smirnov 均匀分布检验，过程如下：

1. 对于 n 个样本 $X_1, X_2, \cdots, X_n$, 写出其经验分布函数 $F_n(X)$
2. 提出假设检验问题：

   H_0: 样本来自的总体服从均匀分布 v.s H_1: 样本来自的总体不服从均匀分布
3. Kolmogorov–Smirnov 统计量为：$D = \max |F_n(X) - F_0(x)|$,$F_0(x)$ 是均匀分布函数.
4. 与 K-S 检验的 D 统计量的临界值作比较, 检验的拒绝域为 $\{D > D_{n,\alpha}\}$

以所有车辆（以 ID 区分）第一次出现的时间作为到达时间，我们对所有的数据文件进行 K-S 检验，显示结果如下:

表 4.1: K-S 检验结果

序号	A 组数据		B 组数据		C 组数据		D 组数据	
	文件	P 值	文件	P 值	文件	P 值	文件	P 值
1	A1.csv	0.2818	B1.csv	0.0035	C1.csv	0.3289	D.csv	0.8295
2	A2.csv	0.8173	B2.csv	0.5168	C2.csv	0.9622	-	-
3	A3.csv	0.0840	B3.csv	0.0731	C3.csv	0.3673	-	-
4	A4.csv	0.9176	B4.csv	0.6134	C4.csv	0.3019	-	-
5	A5.csv	0.7921	B5.csv	0.3247	C5.csv	0.0995	-	-
6	-	-	-	-	C6.csv	0.7414	-	-

在显著性水平 $P = 0.05$ 的条件下，17 组文件只有一组文件不满足均匀分布的假设，这可能是由于一些意外情况导致的。因此，我们认为对于所有的数据，车辆的到达时间上是满足**均匀分布**的。

4.2.2 模型的基本原理和公式

GCD-KS 模型依赖于两个基本原理：

- **原理 1：车辆在路口停下后，再次启动的时刻一定是绿灯开始的时刻:** 车辆在路口停下一定是在等待红灯结束，绿灯亮时车辆启动。由于启动时间和采样造成的误差很小，我们可以这样近似处理。
- **原理 2: 在一个周期内，通过路口的车辆一定集中于绿灯时段通过：** 如果没有闯红灯等违反交通规则的情况，这一点是显然的。

根据原理 1，记车辆 i 启动的时刻为 t_i^{start}, 那么任意两辆车的启动时刻相差一定满足：

$$(t_i^{\text{start}} - t_j^{\text{start}}) \mod C = 0 \tag{4.1}$$

假设数据集一共记录了 n 辆车有**在路口停下后启动**的过程，那么可以将这些时刻两两相减，会得到 $n(n-1)/2$ 个值。用辗转相除法得到这 $n(n-1)/2$ 的最大公约数。这些数落在合理区间（比如 30s——200s）的作为周期的候选，记 $C_{candidate} = (C_1, C_2, \cdots, C_n)$

根据原理 2 以及车辆的到达时间服从均匀分布的条件，我们将车辆通过路口发生的时刻映射到一个周期 $[0, C]$ 上：

$$\tau_i^c = (t_i^c - t_{ref}) \mod C \tag{4.2}$$

其中 t_{ref} 是一个作为起始的参考时刻。这样映射之后，如果 C 是正确的周期，那么在一个周期内，会出现一个明显的现象：车辆通过路口发生的时刻一定集中于绿灯时间段，即不是一个**均匀分布**。反之，如果 C 不是周期，那么根据车辆的到达时间服从均匀分布，车辆穿越路口的时间应该也是一个均匀分布。

因此，只需要对 $C_{candidate} = (C_1, C_2, \cdots, C_n)$ 进行 K-S 检验，选择 P 值最小的那个（即最不可能是均匀分布的）作为信号灯的周期。

4.3 求解绿灯时间的绿色区间估计模型

4.3.1 简单绿色区间估计

模型 1 成功解决了完整周期 C 的求解方法，但是，要知道红灯周期和绿灯周期，我们必须独立地解出一个来。绿色区间估计模型完美地解决了求解绿灯周期的问题，模型的方法如下：

1. 首先分析并记录车辆通过路口的时刻 t_i^c，并将其映射到一个周期上，即:

$$\tau_i^c = (t_i^c - t_{ref}) \mod C \tag{4.3}$$

2. 选择合适的组距（本题宜选择 1s-2s），绘制一个周期的通过事件直方图 h_{cross}

3. 类似的，分析并记录车辆停车的时刻 t_i^c，并将其映射到一个周期上，即:

$$\tau_i^s = (t_i^s - t_{ref}) \mod C \tag{4.4}$$

4. 选择合适的组距（本题宜选择 1s-2s），绘制一个周期的停车事件直方图 h_{stop}

5. 将两张直方图合并在一个图里。

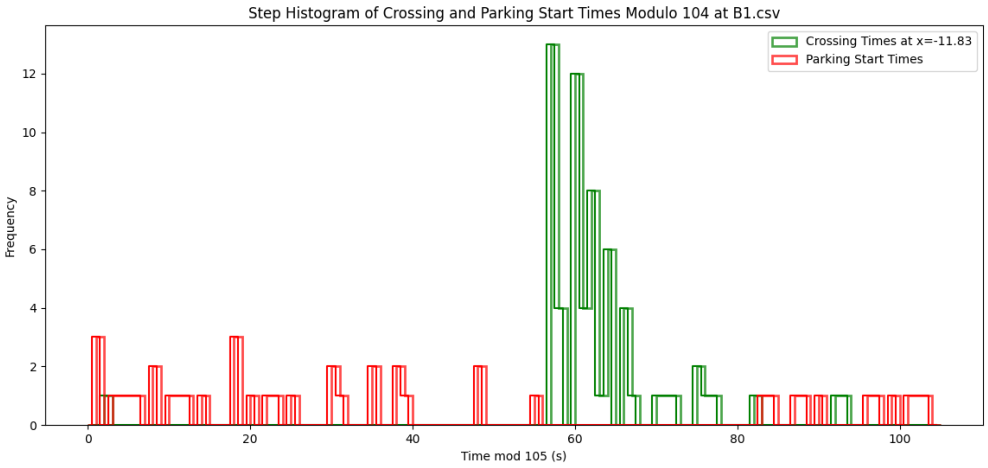


图 4.3: 绿色区间估计的直方图, 横坐标为时刻，纵坐标为频数。

如果选择了合适的周期，且数据没有异常和较大误差的情况下，通过事件和停车事件会出现如上图一样的分布：这是因为，红灯的时候，车辆应该是不能通过路口的，使得通过事件少而停车事件多；而路灯车辆会大量通过路口，使得停车事件少而通过事件多。由于红灯时间和绿灯时间一定是一段连续的时间，这样一来，我们就能根据图形得出初步的结论。

4.3.2 复合绿色区间估计

进一步地，考虑一些矛盾事件（比如同一时间段既有停车事件又有通过事件），我们需要细化上面的模型，以达到更好的估计效果。

- **对停车事件的检查：** 不是所有的停车事件都是由于红灯禁行造成的，比如，离路口较远的停车事件可能是因为上下车等因素造成的, 离路口近的停车事件才越有可能是红灯禁行造成的。因此，我们对不同的停车事件赋予权值 ω，最靠近路口的停车事件 $\omega = 1$，根据远离情况依次赋更小的值，直至为 0. 最后，用这个权值乘上停车频数并向下取整，得到新的停车直方图。

- **对矛盾事件的排除：** 在检查了异常停车事件后，我们需要排除一对矛盾事件——在相同周期秒内既有停车又有通过——这是不符合常理的，因为在路口附近，红灯期间车辆都会停下而绿灯期间大家车辆通过。为此，认为是采样或者轨迹误差造成的，我们忽略通过事件而保留停车事件。

- **得到绿灯区间：** 将通过事件直方图减去加权后的停车直方图，得到新的直方图 h_{green}. 直方图 h_{green} 大于某个阈值（一般是 0 或者 1）的时刻即为绿灯时刻。

$$h_{\text{green}} = h_{cross} - \omega h_{stop}$$

最后，我们给出绿色区间计算的伪代码：

Algorithm 1 计算绿灯区间

Set $green\_start$ to **None**
for $second = 0$ **to** $cycle\_length$ **do**
    if $h_{green}[second] > threshold$ **then** ▷ 如果大于阈值
        if $green\_start$ is **None then** ▷ 绿灯开始
            Set $green\_start$ to $second$
        end if
    else
        if $green\_start$ is not **None then**
            Append $(green\_start, second)$ to $green\_intervals$
            Set $green\_start$ to **None** ▷ 绿灯结束，等待下一个绿灯
        end if
    end if
end for

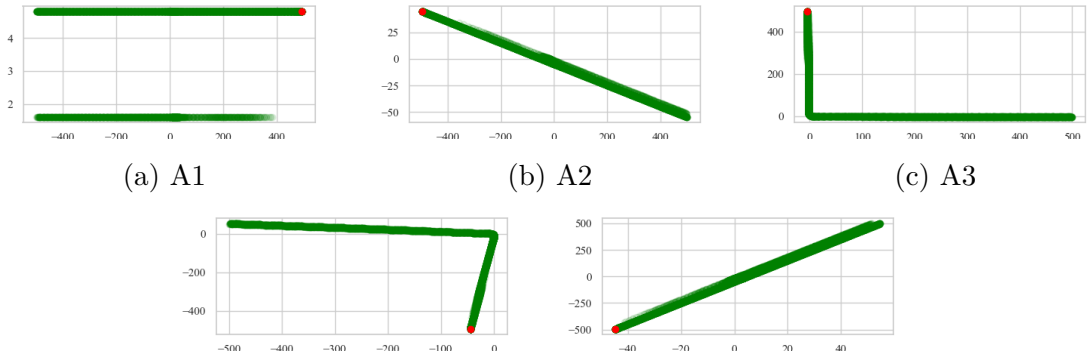
五　问题的求解

5.1　问题 1 的求解

5.1.1　问题 1 数据集的可视化与处理

分析 A1-A5 的数据集发现，各个路口每小时通过的车辆均值为 96。这表明了路口的车辆是相对稀疏的，因此，不同于常见交通流里会遇到的堵塞流和恢复流，这里几乎没有排队现象。

分析 A1-A5 的轨迹图，可以发现记录的车辆都是沿着一条轨迹的：直行或者转弯；但部分车辆存在变道现象，这给确定路口位置和路口通过事件增加了一定的难度。



(a) A1　(b) A2　(c) A3

(d) A4　(e) A5

图 5.4: A1-A5 的轨迹图. 其中红点代表起点

5.1.2 周期 C 的求解

应用 GCD-KS 模型，我们首先需要识别出所有的路口。对于问题 1 的数据集，简单分析坐标的频数之后，会发现车辆在各个路口几乎只有一个停车点，我们把这些停车点作为路口坐标。

其次，我们需要识别出所有跨越路口的事件，即识别出所有的 t_i^C 对于各个路口的所有车辆，以及所有在路口停车后启动的车辆 t_{start}^i.

对于每个路口，找出最大公约数作为周期的候选，接着通过假设检验，选取 P 值最小的作为每个路口的周期，得到的结果如下：

表 5.2: 每个路口最佳估计周期与其对应的 P 值

路口	**最佳估计周期**	**P-Value**
A1	105	4.67×10^{-13}
A2	88	4.20×10^{-8}
A3	105	3.40×10^{-8}
A4	88	4.34×10^{-11}
A5	88	7.39×10^{-7}

注：虽然有些 P 值最小，但是与差分结果差距较大，我们综合选择了相对最小的 P 值对应的周期。

5.1.3 绿灯周期 C_G 的求解

应用绿色区间估计，首先需要求出路口通过事件和路口停车事件。路口通过事件的时刻可以用车辆穿越路口点前后的时间均值代替，而路口停车事件的检测只需要判断前后位置是否几乎不变即可。接着，画出 A1-A5 的通过-停车事件直方图如下

可以看到，绿色的通过事件总是在绿灯区间里连续、高频出现，而红色的停车事件在红灯周期内均匀分布，我们只需要根据绿色直方图的分布和绿色区间长度求解绿灯时长即可。由于车辆的稀疏性，可能会导致一段时间内既无停车事件又无通过事件，我们需要根据图中的分布规律自行延长绿灯时间。

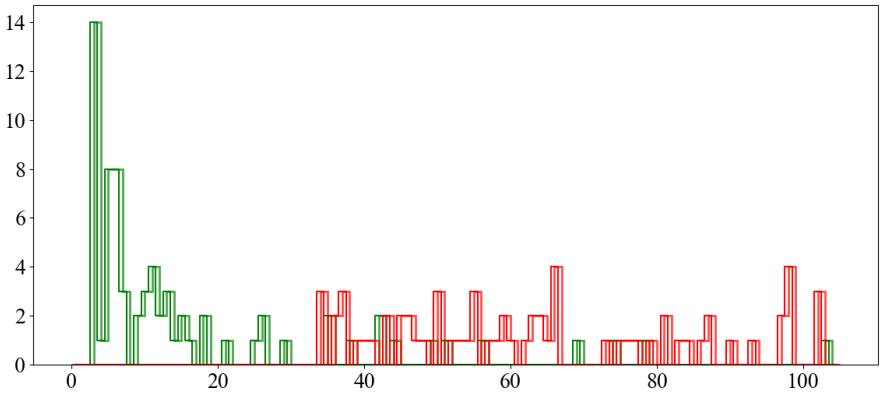


图 5.5: A1 路口的绿色区间直方图

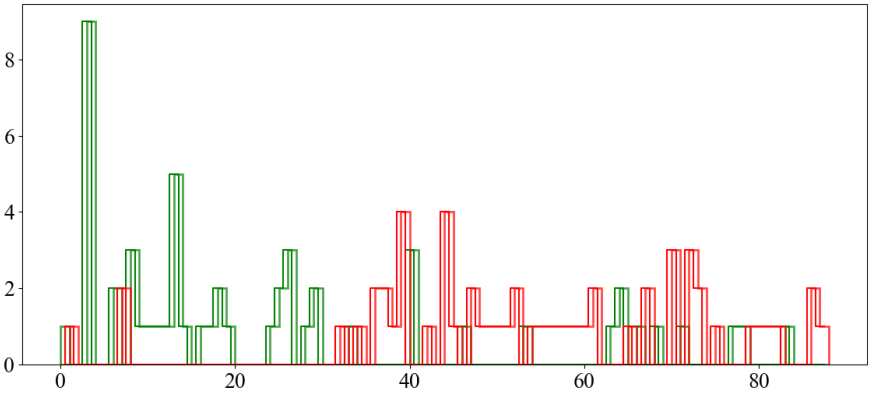


图 5.6: A2 路口的绿色区间直方图

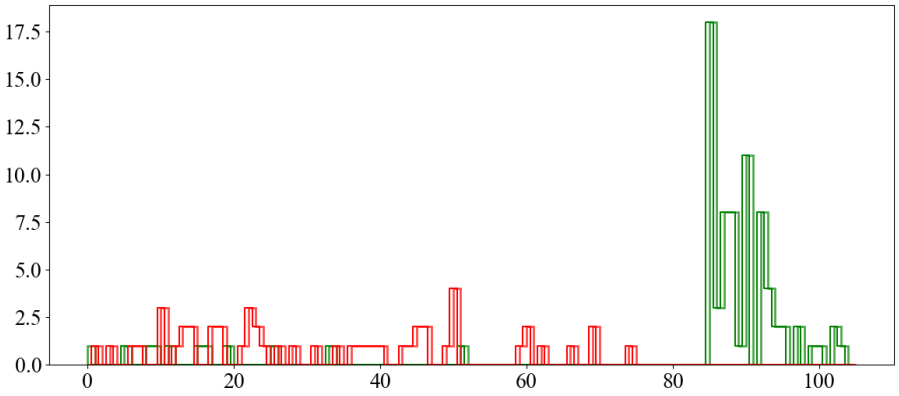


图 5.7: A3 路口的绿色区间直方图

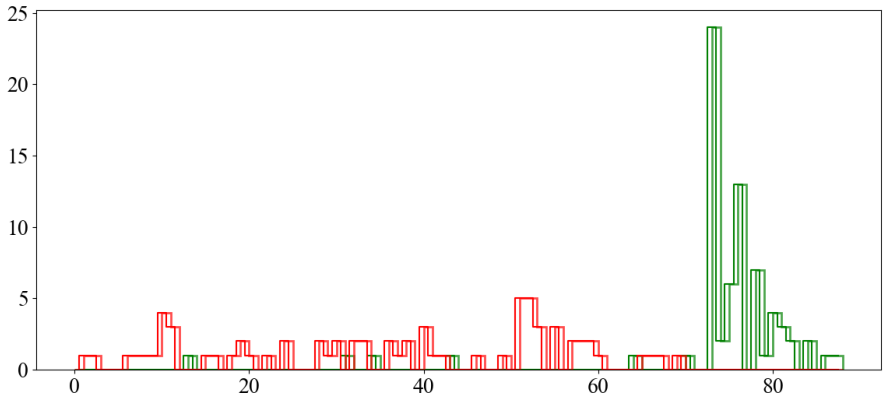


图 5.8: A4 路口的绿色区间直方图

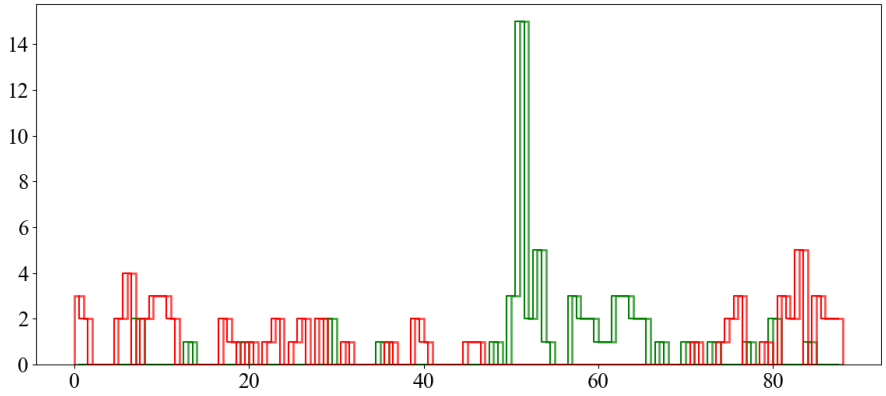


图 5.9: A5 路口的红绿灯直方图

5.1.4 结论

根据以上的求解过程，我们得到了路口 A1-A5 信号灯的红绿周期，按要求填入表 1 后如下表所示：

表 5.3: 路口 A1-A5 各自一个方向信号灯的绿灯时间长度

路口	A1	A2	A3	A4	A5
红灯时长（秒）	75	60	82	69	64
绿灯时长（秒）	30	28	23	19	24

5.2 问题 2 的求解

5.2.1 问题 2 数据集的分析与处理

与问题 1 相比，问题 2 的数据量明显地少于问题 1，B1-B5 各个路口每小时通过的车辆均值仅为 54，这表明了确实只获取了部分样本车辆的行车轨迹。因此，这可能会成为影响我们模型精度的一个重要因素。而问题 2 的中 B1-B5 的轨迹图与上一问数据的轨迹图几乎一样，因此就不再给出。

考虑到问题 2 相对于问题 1，存在多个停车点，这对于识别路口位置造成了一定影响。我们通过 DBSCAN 密度聚类算法，先提取停车点，再对停车点的坐标进行密度聚类，以密度最高的点作为路口点，得到的结果如下:

路口	B1	B2	B3	B4	B5
x	-11.4	-3.64	11.51	4.8	-18.97
y	-1.6	11.83	0.46	-11.4	0.29

表 5.4: 最佳路口位置表格

5.2.2 周期 C 与绿灯周期 C_G 的求解

再次运用 GCD-KS 模型，这里直接给出周期 C 的求解结果：

表 5.5: B 系列的估计周期长度和最小 P 值

路口	**最佳估计周期长度**	**P-Value**
B1	105	8.05×10^{-21}
B2	116	1.42×10^{-8}
B3	88	1.71×10^{-4}
B4	105	1.09×10^{-8}
B5	116	4.77×10^{-3}

注：虽然有些 P 值最小，但是与差分结果差距较大，我们综合选择了相对最小的 P 值对应的周期。

同样，再次运用绿色区间估计，这里直接给出绿色区间直方图的结果：

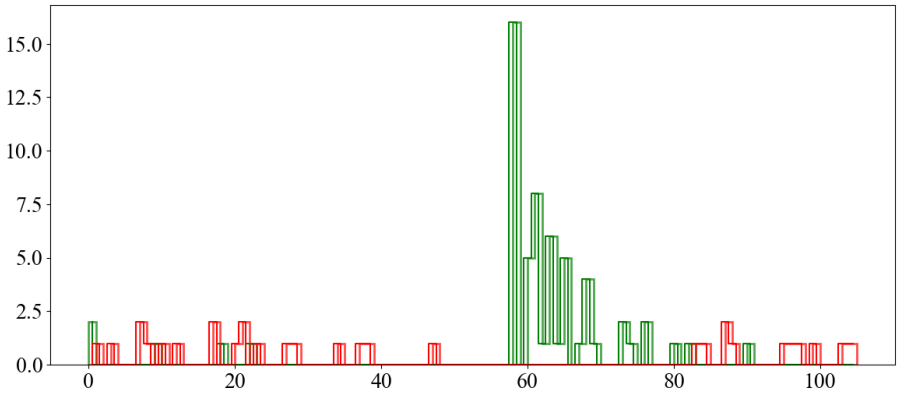


图 5.10: B1 路口的绿色区间直方图

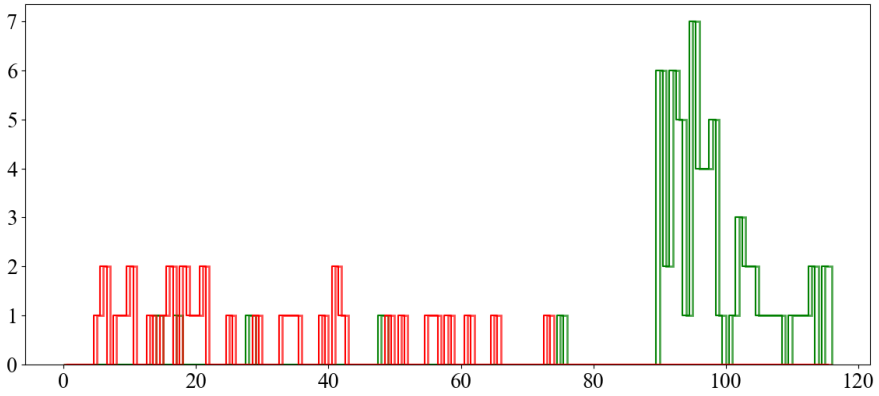


图 5.11: B2 路口的绿色区间直方图

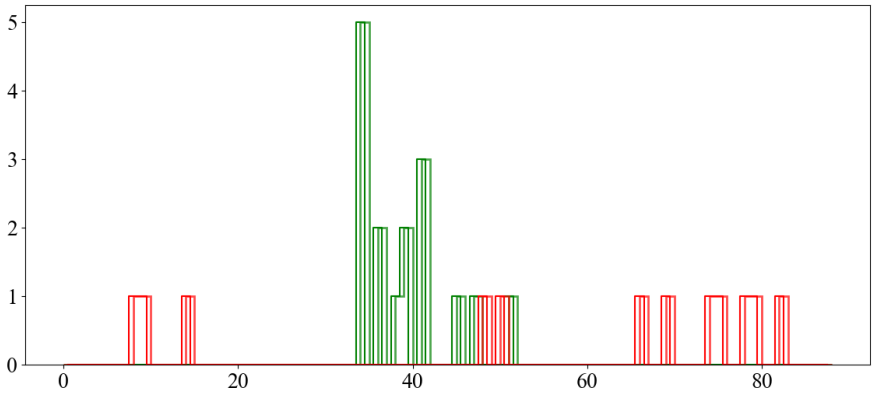


图 5.12: B3 路口的绿色区间直方图

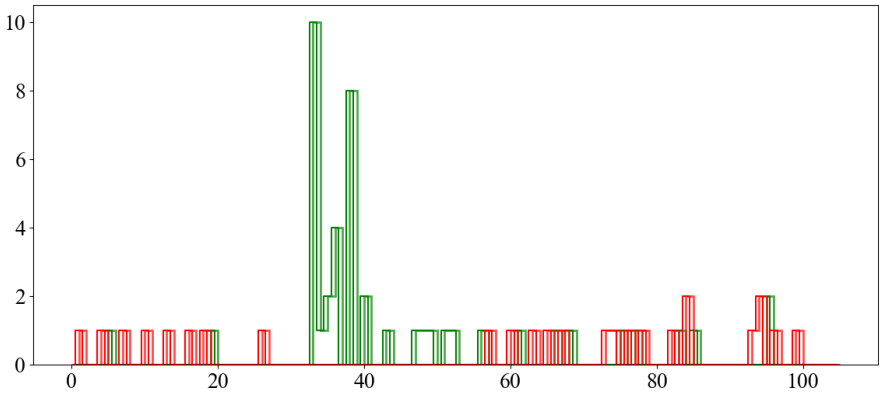


图 5.13: B4 路口的绿色区间直方图

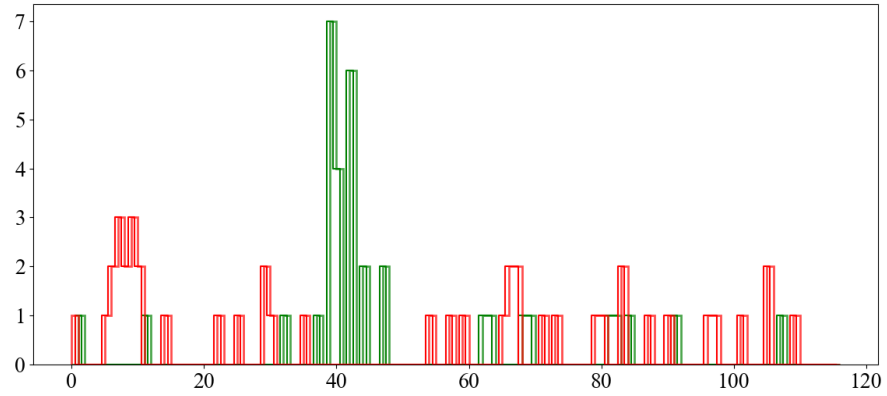


图 5.14: B5 路口的红绿灯直方图

5.2.3 结论

根据以上的求解过程，我们得到了路口 B1-B5 信号灯的红绿周期，按要求填入表 1 后如下表所示：

表 5.6: 路口 B1-B5 各自一个方向信号灯的绿灯时间长度

路口	B1	B2	B3	B4	B5
红灯时长（秒）	79	89	73	80	99
绿灯时长（秒）	26	27	15	25	17

5.2.4 灵敏度分析

1. **采样比例对测量结果的影响:** 对于红绿灯周期，采样比例的提升将直接提高测量结果的准确性。当采样比例较低时，可能会出现多个周期内仅有单一车辆通过的

情况，这会导致测量结果偏向周期的整数倍，从而降低测量精度。在处理红绿灯配比时，若将多个周期的数据合并至单一周期内进行分析，车辆数据的稀疏性可能导致绿色时间区间的直方图出现大量数据缺失，进一步影响精度。

2. **车流量对测量结果的影响：** 在考虑小车流量的场景下，增加车流量对模型的影响较小。在周期计算中，模型主要采用路口最前端车辆的行驶轨迹数据，因此车流量的增加对周期测量影响不大。然而，在计算红绿灯时长时，车流量的增加可能导致车辆在路口处聚集，虽然模型通过分析车辆的停留时间来估计红灯时长，但由于主要考量的是车辆通过红绿灯的瞬间时间，所以车流量对红绿灯时长测量的精度影响较小。

3. **定位误差对测量结果的影响：** 定位误差中，小于单位时间内移动 0.1 的误差通常会被过滤掉，对结果无明显影响。关于信号波动导致的定位点波动，例如车辆在特定时刻的位置突变或倒退等，我们会对这类数据进行筛选处理。

5.3 问题 3 的求解

问题 3 是检测两小时内同一路口红绿灯周期是否发生了变化。为此，我们需要先检验完整周期是否发生了变化，再检验红绿灯周期的变化。

5.3.1 问题 3 的数据集概览

问题 3 与问题 1、2 不同的是，数据集的时间从 1 小时延长到 2 小时，且各个路口每小时通过的车辆均值为 78，这介于问题 1 的均值和问题 2 的均值。对问题 3 绘制散点图后，发现轨迹同问题 1、问题 2 类似。且停车地点个数介于问题 1、2 之间，因此，数据集的处理不是问题 3 的重点。

5.3.2 完整周期的检验

为了检验完整周期是否发生了变化，我们采用了两种方法：

- **利用启动时刻的差分：** 我们计算了相邻启动时刻的差分 $t_{start}^{i+1} - t_{start}^{i}$，结果显示，差分结果高度一致，这有理由说明完整周期在整个过程中是不变的。

- **滑动窗口的 GCD-KS 模型：** 我们采用滑动窗口的办法，计算每一段 GCD-KS 模型得出的周期，结果显示，每一段得出的周期几乎都是相同的。在本题中滑动窗口大小为 1000s, 步长为 100s。

表 5.7: 启动时刻的差分结果

路口名	差分 1	差分 2	差分 3	差分 4	差分 5	差分 6
C1	88	88	88	88	88	264
C2	88	440	880	176	528	352
C3	105	1	104	105	105	105
C4	105	210	105	105	105	105
C5	88	176	176	176	616	176
C6	525	315	315	105	945	105

由于篇幅有限，我们按顺序展示了部分差分结果，完整的差分结果见支撑材料。

表 5.8: 滑动窗口的 GCD-KS 模型结果

路口名	C1	C2	C3	C4	C5	C6
出现最多的周期值	89	88	104	105	88	105

由于篇幅有限，我们仅展示了频数最高的几个周期值，完整的周期值见支撑材料。

综合以上，由于移动窗口可能受窗口大小影响，周期可能存在一定偏差，最终得出的结论是：

路口名	C1	C2	C3	C4	C5	C6
周期	88	88	105	105	88	105

表 5.9: C1-C6 的估计周期长度

5.3.3 红路灯占比变化的识别

由于红灯时间和车辆的停车时间是高度相关的：一般情况下，红灯时间越长，停车时间会显著增加；红灯时间越短，停车时间则会显著减少。因此，我们只要检测停车时间的序列是否存在变点即可。

在这里考虑到样本量较少导致的停车时间数据可能有较强的突变性，我们采用了差分平均的方法寻求变点，具体步骤如下：

1. 对于给定的停车序列 $P = \{\Delta t_{p1}, \Delta t_{p2}, \cdots, \Delta t_{pn}\}$, 窗口大小为 w 的移动平均是：

$$MA_t = \frac{1}{\omega} \sum_{i=t-w+1}^{t} \Delta t_{pi} \tag{5.5}$$

2. 为了检测数据中的变化点，我们计算移动平均的一阶差分:

$$\Delta MA_t = MA_t - MA_{t-1} \tag{5.6}$$

3. 我们检测移动平均差分的绝对值中的峰值来识别潜在的变点。具体来说，如果某个差分的绝对值显著高于其他值，这可能表明在该点附近数据的行为发生了显著变化。

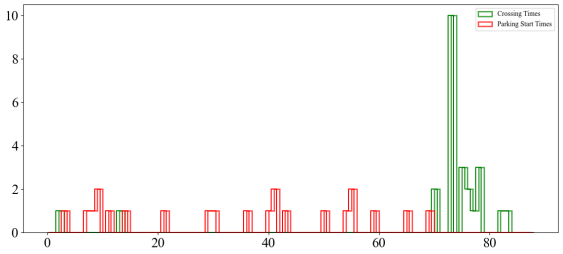
在我们是数据中，我们应用了 10、15 两种窗口大小，对 C1-C6 数据集分别检测变点，得到的结果如下：

表 5.10: 移动平均差分检验变点的结果

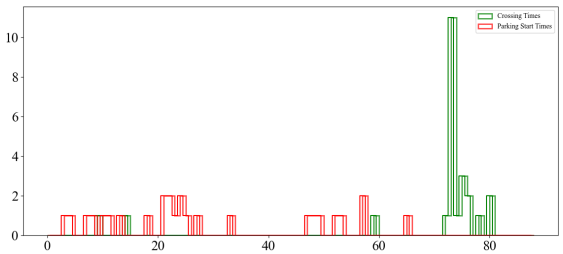
路口名	窗口大小	变点索引	左侧均值	右侧均值	均值改变	变点时间
C1	10	52	22.02	17.79	4.23	3587
C2	10	35	28.8	34.97	−6.17	3567
C3	10	71	36.15	33.9	2.25	3741
C4	15	27	41.56	38	3.56	1699
C5	15	57	22.42	18.21	4.21	5880
C6	10	61	34.8	39.8	−5	6170

5.3.4 红绿灯占比变化的求解

求解出了红绿灯占比变化的时间点，可以以该时间点为界，左右各自构造一个绿色区间直方图，通过求解绿色区间进而求出绿灯时间的改变，来反推红灯时间的改变。



(a) C2 变化前



(b) C2 变化后

图 5.15: 红绿灯周期的变化：以 C2 为例

我们绘制出了所有绿色区间直方图，求解了变化前后的绿色区间，并剔除了周期变化小于 3s 的结果（认为其不发生变化），得出的结果如下：

表 5.11: 路口绿灯时间的改变

路口名	C1	C2	C3	C4	C5	C6
原绿灯时间	35	15	23	22	31	27
改变后绿灯时间	38	10	23	23	33	23

这里我们认为 C3、C4、C5 的变化幅度太小，不符合切换的一般规律，认定其不发生红绿周期切换。

5.3.5 结论

根据以上的求解过程，我们得到了路口 C1-C6 信号灯的红绿周期以及切换时刻，按要求填入表 3 后如下表所示：

表 5.12: 路口 C1-C6 各自一个方向信号灯的绿灯时间长度

路口	C1	C2	C3	C4	C5	C6
周期 1 红灯时长（秒）	53	73	82	82	66	78
周期 1 绿灯时长（秒）	35	15	23	23	22	27
周期切换时刻	3587	3567	无	无	无	6170
周期 2 红灯时长（秒）	50	78	-	-	-	82
周期 2 绿灯时长（秒）	38	10	-	-	-	23 -

5.4 问题 4 的求解

5.4.1 问题 4 数据集的分析与可视化

问题 4 的数据集给出了一个路口所有方向的车辆轨迹，根据十字路口的构造以及所给出的数据，在一个方向上，车辆会有直行、左转和右转三种轨迹，四个方向共 12 种轨迹。我们使用 K-means 算法对车辆的轨迹数据进行聚类，得到不同行驶方向的车辆轨迹, 每个方向是 3 种轨迹，一共 12 种轨迹

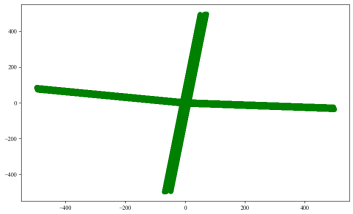
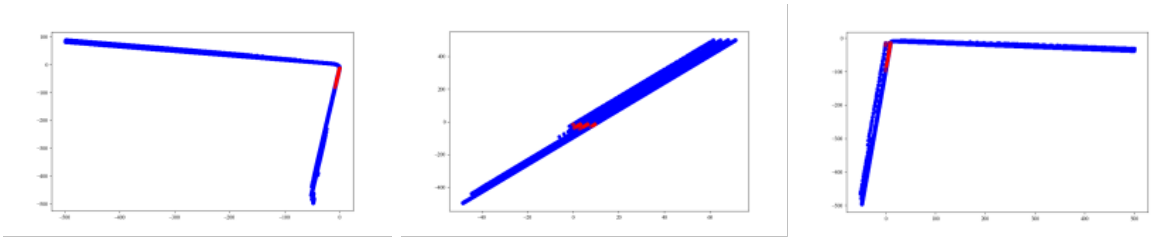


图 5.16: 原始轨迹的叠加



(a) 一个方向的左转轨迹　(b) 一个方向的直行轨迹　(c) 一个方向的右转轨迹

图 5.17: 聚类结果

5.4.2 路口周期的求解

首先我们通过 GCD-KS 模型对这 12 个聚类结果分别求周期，求得的周期都为 142s，说明我们的聚类效果非常好。接下来，我们需要从这一个完整周期里求解出十字路口的红绿周期。实际上，十字路口的运行模式是一个状态机，一共有四种模式：

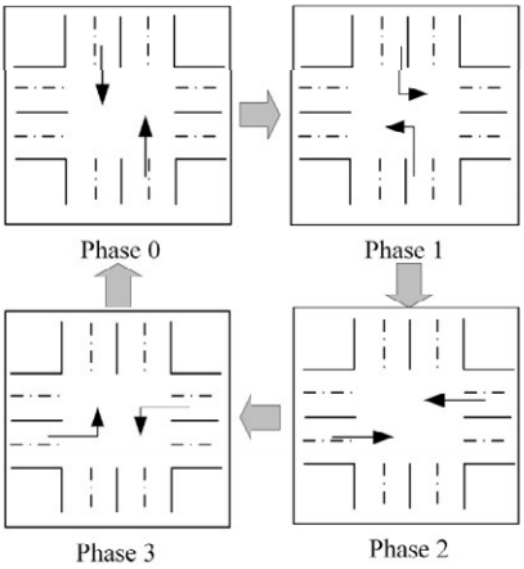


图 5.18: 十字路口的运行模式

然后，我们对每个方向的轨迹进行绿色区间估计，画出每个方向的绿色区间直方图，进而得到每一类数据的绿色区间如下表所示：

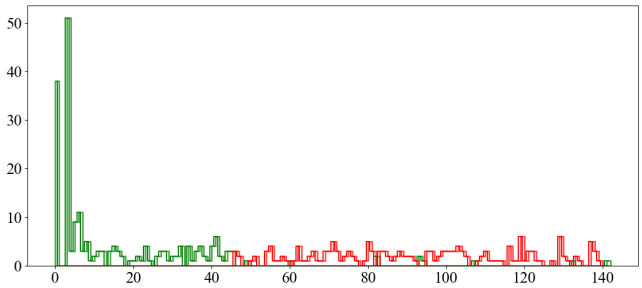


图 5.19: 例：十字路口上下来向车辆直行的绿色区间直方图

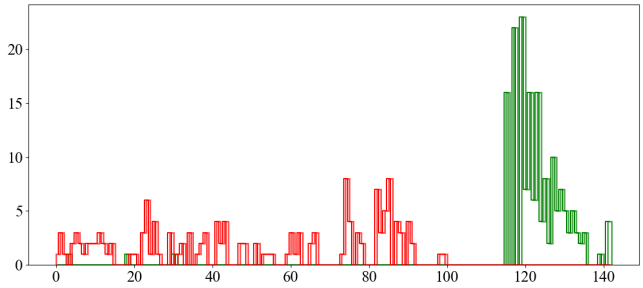


图 5.20: 例：十字路口左右来向车辆左拐的绿色区间直方图

表 5.13: 绿灯信号时间表

方向	时间（秒）	
	开始	结束
上直行	0	46
上左转	49	70
上右转	0	51
下直行	0	47
下右转	0	55
下左转	46	70
右右转	74	123
右左转	112	142
右直行	78	112
左左转	114	142
左右转	76	117
左直行	68	112

接下来，我们使用通过全向联合周期确定法来确定路口信号灯的周期，这种方法利用了多个方向的不同信号灯变化时间，其一般步骤是:

- **Step1：**确定不同方向不同轨迹红绿灯的在周期内的相位，找到红绿灯切换的时间点 ti。
- **Step2：**确定该路口统共有几种状态（通常情况下，十字路口有四种状态，T 字路口有三种状态），并根据状态判断不同红绿灯切换时间点属于哪两个状态的临界点 $ti \in T$。
- **Step3：**状态的切换点取该状态取所有红绿灯切换时间点的均值（四舍五入取整数）。接着，根据十字路口运行的四种模式，我们将这 12 个绿灯时间分配到四个周期里，并用算术平均值代替每个周期的绿灯时间，得到以下表格:

表 5.14: 分配到不同阶段的时间点

起始时刻	phase0	phase1	phase2	phase3
0	46	70	123	142
0	49	70	112	142
0	51	74	112	142
0	47	78	114	142
0	55	76	117	142
0	46	68	112	142
均值	49	73	115	142

5.4.3 结论

取上表的均值作为路口周期划分的最终结果，如下表所示：

阶段	phase0	phase1	phase2	phase3
时间	49	24	42	27

表 5.15: 路口信号灯的周期划分

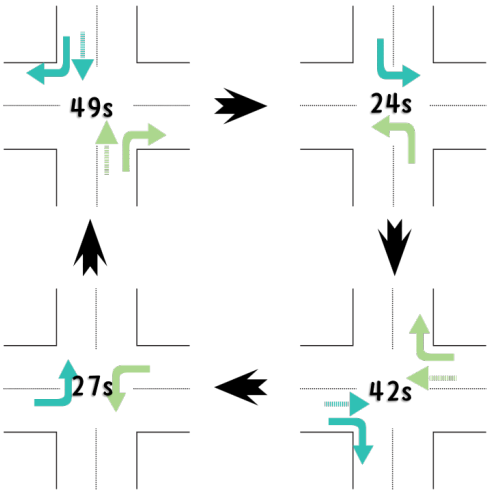


图 5.21: 周期轮换示意图

六　模型的优缺点分析

本题主要运用到了两个基本模型，一个是 GCD-KS 模型，一个是绿色区间估计模型，这两个模型对于分析和处理交通信号灯周期问题十分关键，下面主要是对这两个模型的分析

6.1 模型的优点

在估计交通信号灯周期方面，该模型具有如下的优点。

- **优点 1 · 模型的误差较小：**在对周期的估计种，GCD-KS 模型对误差十分敏感，1s 左右的误差就能引起 P 值的急剧上升。在对绿灯时间的估计中，将停车事件和启动时间综合考虑，并投影在一个周期内，提高了数据的利用率，减小了误差。
- **优点 2 · 模型的鲁棒性好：**车流密度和选择性记录对于模型的影响较小，因为模型关注的是车辆通过路口的时刻，其主要受交通信号灯影响较大。
- **优点 3 · 模型的适用性广：**模型的限定条件较少，适用的场景广泛。

6.2 模型的缺点与改进

与此同时，该模型也具有如下的缺点。

- **缺点 1 · GCD-KS 模型对长绿灯时间的效果较差：**如果在一个完整周期内，大部分时间都是绿灯，假设检验的结果可能总是呈现均匀分布，这提高了假阳性的概率。尽管现实情况下很难遇到，但这仍是模型的一大缺点
- **缺点 2 · 不适用于交通模式高度变化的环境：**如果交通模式改变较快，那么投影到一个周期的方法可能就丧失了周期共性，导致对信号灯的观察产生一定的误差。

同时，在这里给出进一步优化模型的思路：

- **优化假设检验的方法：**本文使用的假设检验方法是非参数的 $K-S$ 检验方法，这一假设检验对数据量和数据稳定性具有一定要求，可以尝试其他假设检验方法，以优化模型
- **优化对停车事件的判断：**实际上，在数据集中，出现了少数在路口异常停车的情况（比如在路口出现倒退情况），优化的方向可以尝试解释并处理这些情况，而不是简单的剔除。
- **尝试更好解决周期变化问题的方法：**实际上，本文针对周期变化问题，采用了一种非常简单的移动平均差分的方法去进行比较，但是，这种方法仍然是比较粗糙的。考虑到数据量较少的情况，可以尝试更好解决周期变化问题的方法。

附录

支撑材料的文件架构

其中，B 题源码每个文件夹都包含了对每个数据集（A,B,C,D）处理和计算的所有源码。

LaTex 源码是生成本 PDF 文件的 LaTex 源代码

支撑论文是本文的重要参考文献

重要的一些代码

计算停车区间的函数

```
def find_parking_intervals(vehicle_data):
    vehicle_data['parked'] = (vehicle_data['x'].diff().eq(
        0) & vehicle_data['y'].diff().eq(0))
    vehicle_data['parking_block'] = (~vehicle_data['parked']).cumsum()
    parking_intervals = vehicle_data[vehicle_data['parked']].groupby('
        parking_block').agg(
        start_time=('time', 'min'),
        end_time=('time', 'max'),
        # Capture the last x position at the end of the interval
        last_x=('x', 'last'),
        # Capture the last y position at the end of the interval
        last_y=('y', 'last')
    )
    parking_intervals['duration'] = parking_intervals['end_time'] - \
        parking_intervals['start_time'] + 1
    parking_intervals = parking_intervals[parking_intervals['duration'] >
         1]
    return parking_intervals[['start_time', 'end_time', 'duration', '
        last_x', 'last_y']]
```

得出最大公约数空间通过 K-S 检验搜索最优解函数

```
def mfagcd(tg, delta, cmin, cmax):
    n = len(tg)
    if n < 4:
        return 0
    C = set()

    # Iterating over each unique combination of three different tg values
    for j in range(n-2):
        for k in range(j+1, n-2):
            for h in range(k+1, n):
                # Considering error bounds for each tg value
                t1_options = np.arange(tg[j] - delta, tg[j] + delta + 1)
                t2_options = np.arange(tg[k] - delta, tg[k] + delta + 1)
                t3_options = np.arange(tg[h] - delta, tg[h] + delta + 1)

                # Calculate GCD for each combination of tg values within
                    the error bounds
                for t1 in t1_options:
                    for t2 in t2_options:
                        for t3 in t3_options:
                            cycle = gcd(t3 - t1, t2 - t1)
                            if cmin <= cycle <= cmax:
                                C.add(cycle)

    # Finding the most common cycle length in the set of viable cycles
    if C:
        cycle_counts = Counter(C)
        return cycle_counts
    else:
        return set(0)


delta = 1
cmin = 30
cmax = 240
```

绿色区间估计模型

```
    def combined_green_interval_estimation(crossing_histogram,
        stop_histogram, stop_distances, cycle_length):
        # 根据距离计算权重
        weights = np.where((stop_distances > 0) & (stop_distances <= 10),
            1,np.where((stop_distances > 10) & (stop_distances <= 20),
            0.4, np.where((stop_distances > 20) & (stop_distances <= 30),
            0.2, 0)))
    # 应用权重到停车直方图
        weighted_stops = stop_histogram * weights

    # 计算剩余的交叉事件直方图
        rem_histogram = crossing_histogram - weighted_stops

    # 阈值设置为1
        threshold = 1

    # 确定绿灯区间
        green_intervals = []
        green_start = None
        for second in range(cycle_length):
            if rem_histogram[second] >= threshold:
                if green_start is None: # 绿灯开始
                    green_start = second
            else:
                if green_start is not None: # 绿灯结束
                    green_intervals.append((green_start, second))
                    green_start = None
    # 如果周期结束时仍在绿灯状态
        if green_start is not None:
            green_intervals.append((green_start, cycle_length))

        return green_intervals

    grouped_by_vehicle = data.groupby('vehicle_id')
    crossing_data = []
```

滑动窗口 GCD-KS 模型

```
    for i in range(1, 7):
    data = pd.read_csv(f'C{i}.csv')
    grouped_data = data.groupby('vehicle_id').apply(
        find_parking_intervals)
    posx, posy = grouped_data[['last_x', 'last_y']].value_counts().idxmax
        ()

    final_data = grouped_data[(grouped_data['last_x'] == posx) & (
        grouped_data['last_y'] == posy)]

    window_size = 1500
    step_size = 100
    best_p = 1
    best_cycle = 0
    best_cycle_counts = {}

    for start_time in range(0, 7200, step_size):
        end_time = start_time + window_size
        window_data = final_data[(final_data['end_time'] >= start_time) &
            (final_data['end_time'] < end_time)]
        tg = window_data['end_time'].unique()
        estimated_cycle = GCD(tg) # 假设 GCD 函数返回周期估计值

        # 进行KS检验
        for cycle_length in estimated_cycle:
            crossing_times_cur = window_data[(window_data['end_time'] >=
                start_time) & (window_data['end_time'] < end_time)]
            adjusted_times = crossing_times_cur['end_time'] % cycle_length
            _, p_value_ks = stats.kstest(adjusted_times, 'uniform', args
                =(0, cycle_length))

            if p_value_ks < best_p:
                best_p = p_value_ks
                best_cycle = cycle_length

            # 更新最佳周期计数
            best_cycle_counts[best_cycle] = best_cycle_counts.get(
                best_cycle, 0) + 1
```

参考文献

[1] AXER S. Estimating Traffic Signal States by Exploiting Sparse Low-Frequency Floating Car Data[D]. Dissertation, Braunschweig, Technische Universität Braunschweig, 2017, 2017.

[2] CHUANG Y T, YI C W, TSENG Y C, et al. Discovering phase timing information of traffic light systems by stop-go shockwaves[J]. IEEE Transactions on Mobile Computing, 2014, 14(1): 58-71.

[3] KERPER M, WEWETZER C, SASSE A, et al. Learning traffic light phase schedules from velocity profiles in the cloud[C]//2012 5th International Conference on New Technologies, Mobility and Security (NTMS). 2012: 1-5.

[4] YU J, LU P. Learning traffic signal phase and timing information from low-sampling rate taxi GPS trajectories[J]. Knowledge-Based Systems, 2016, 110: 275-292.