

# 贵州省数据大赛（初赛）赛题：智能红绿灯

## 一. 大赛背景

现有的城市红绿灯控制系统虽然在一定程度上可以满足指挥路口交通的需要，但是随着城市车辆的增长，城市拥堵情况越来越严重，原有的红绿灯控制系统已经表现出明显的缺点：红绿灯时间相对固定，不能伴随车流量的改变而调整红绿灯的显示时间。我们常常会遇到这样的情况，在一个十字路口，本车道非常拥堵，而垂直车道非常空闲，但是红绿灯仍然不能优先照顾已十分拥堵的车道，只能机械的变换。

本次大赛，利用贵州省贵阳市南明区交通流量数据，包括：公交车 GPS 信息、出租车 GPS 信息、结合高德公司普通市民导航数据，在充分脱敏与保护用户隐私的前提下，模拟贵阳市整体的十字路口交通情况。希望研究者、学生，可以利用这些交通大数据，对贵阳市红绿灯控制系统进行算法建模，根据交通流量情况实时控制红绿灯的亮灯策略，以最大程度的减少拥堵，加快通行速度。

## 二. 智能红绿灯大赛定义

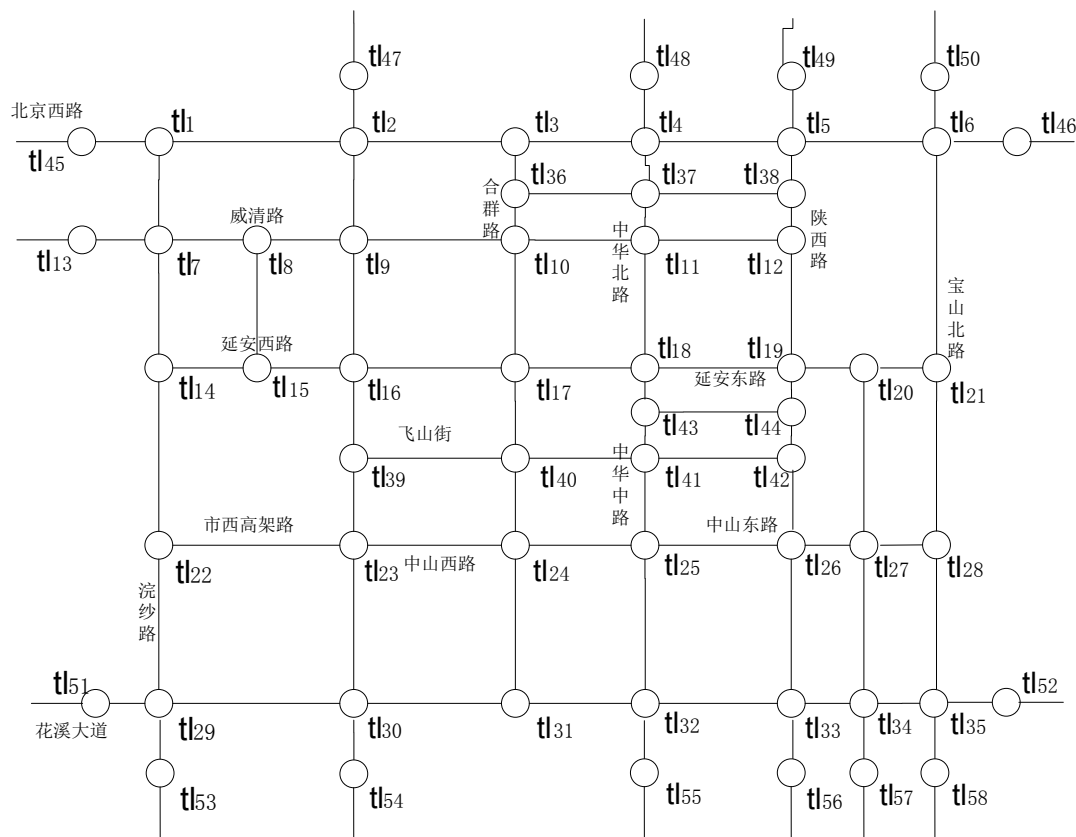


Figure 1. 贵阳市南明区部分区域路口简化示意图

图一是贵阳市南明区部分区域的简化道路与红绿灯位置图，其中红绿灯用  $tl_i$  来表示。

### 2.1 问题简化

说明：该问题的简化方法，在复赛时会有变化，具体请见复赛文档。

1. 为了简化问题，红与绿灯的初始亮灯时间长度都设为单位时间  $T$ ，设置红  
绿灯的时间，必须是  $T$  的倍数。
2. 只有红灯和绿灯，不考虑黄灯。
3. 不考虑车辆掉头情况，只有直行、左转、右转等三种情况。
4. 在一个路口，直行、左转和右转的概率假设为全局固定值。

5. 红绿灯位置图中，只考虑十字路口与 T 字路口情况，不考虑其他情况。
6. 任意两个红绿灯之间的通行距离都是相等的，只需要一个  $T$  时间即可从一个红绿灯到达下一个相邻的红绿灯。

## 2.2 问题定义

给定贵阳市南明区红绿灯信息四元组数据： $G = ( \text{TrafficLight}, \text{TrafficLightStatus}(t_x), \text{TrafficFlow}(t_x), \text{VehicleThroughRate} )$ 。其中  $\text{TrafficLight}$  是红绿灯节点关系， $\text{TrafficLightStatus}(t_x)$  是红绿灯状态， $\text{TrafficFlow}(t_x)$  是车流量信息， $\text{VehicleThroughRate}$  是红绿灯通过率。设定 30 秒为单位时间  $T$ ，将一天中所关心的时间分为  $n$  个  $T$ ， $x \in [1, n]$ ，其中  $|t_x| = T$ 。 $\alpha, \beta, \gamma$  是车辆在红绿灯口，直行、左转和右转的概率， $\alpha + \beta + \gamma = 1$ 。

训练过程：

给定贵阳市红绿灯位置与联接情况:  $\text{TrafficLight}$

提供  $\text{TrafficFlow}(t_x), x \in [1, n]$ ，对于训练数据，我们会给出多天交通流量。

模拟测试过程：

整个测试过程将以模拟器模拟的方式进行。参赛队需要提供实现指定接口的 jar 包。模拟器将输入初始状态的  $\text{TrafficLightStatus}$ ，和初始状态的  $\text{TrafficFlow}$ ，在之后每一个 UT，参赛者的模型必须输出全局红绿灯状态表，模拟器根据该输出，模拟整个贵阳市的交通情况，并计算  $\text{penalty}$ 。为了贴近真实情况，每一个小时，我们将清空中间状态，并输入新的初始状态，要求重新模拟。并将所有的模拟中  $\text{penalty}$  的总和作为最终成绩(排行榜展示  $\text{penalty}$  的变换后的分数)。

限制条件：

模拟过程中会检查全局红绿灯状态表是否满足限制条件，包括红绿灯交规限制与公平性限制，如不满足，将被惩罚一定的分数。

评价方法：

在测试时间段内，车辆在红绿灯前等待的时间总和 (  $\text{Penalty}$  )。取最小者为胜。

### 三. 数据结构描述

#### 3.1 红绿灯位置图关系表

红绿灯位置图可以表示成如下表格，traffic\_light\_table:

当前红绿灯编号 ( TrafficLightID )	来源节点编号 ( FromID )	左转红绿灯编号 ( leftTargetID )	右转红绿灯编号 ( rightTargetID )	直行红绿灯编号 ( straightTargetID )
tl <sub>4</sub>	tl <sub>1</sub>	tl <sub>2</sub>	tl <sub>10</sub>	tl <sub>5</sub>
tl <sub>4</sub>	tl <sub>2</sub>	tl <sub>5</sub>	tl <sub>1</sub>	tl <sub>10</sub>
tl <sub>4</sub>	tl <sub>5</sub>	tl <sub>10</sub>	tl <sub>2</sub>	#

注 1：traffic\_light\_table 是静态表，例子中第一行表示，从 tl<sub>1</sub> 方向到当前红绿灯 tl<sub>4</sub>，左转可以到 tl<sub>2</sub>，右转可以到 tl<sub>10</sub>，直行可以到 tl<sub>5</sub>。

注 2：当值为空时（#），表明是一个 T 字路口，无法到达部分方向。

#### 3.2 当前红绿灯状态表

当前红绿灯状态(t<sub>0</sub>时刻)可以由如下表表示，traffic\_light\_status\_table：

红绿灯编号 ( TrafficLightID )	来源节点编号 ( FromID )	左转红绿灯状态 ( leftLightStatus )	右转红绿灯状态 ( rightLightStatus )	直行红绿灯状态 ( straightLightStatus )
tl <sub>4</sub>	tl <sub>1</sub>	0（注：红灯）	0	1（注：绿灯）
tl <sub>4</sub>	tl <sub>2</sub>	0	1	0
tl <sub>4</sub>	tl <sub>5</sub>	-1	0	1

注 1：traffic\_light\_status\_table 是与时间相关的动态表，例子中第一行表示，在 t<sub>0</sub> 时刻，从 tl<sub>1</sub> 方向到当前红绿灯 tl<sub>4</sub>，左转红绿灯为红灯，不可通行；右转红绿灯为红灯，不可通行；直行红绿灯为绿灯，可以通行。

注 2：结合 traffic\_light\_status\_table 与 traffic\_light\_table，可以得知在某一个时刻，从一个方向到某个红绿灯，是否可以通过，达到另外一个方向。

注 3：当值为-1 时，表明是一个 T 字路口，没有部分方向红绿灯。

### 3.3 交通流量情况表

交通流量情况，可以由如下表表示，traffic\_flow\_table：

红绿灯编号 ( TrafficLightID )	来源节点编号 ( FromID )	车流量 ( traffic_flow )
tl <sub>4</sub>	tl <sub>1</sub>	[10,8,5,0,0,23,13,...,23]
tl <sub>4</sub>	tl <sub>2</sub>	[23,15,12,12,9,9,...,9]

注 1：为简化期间，我们把时间离散化，把 30 秒作为一个时间单位  $T$ ，一天 14 个小时（上午 6 点至下午 8 点）可以分为 1680 时间单位。而一次红绿灯亮灯的最小时间段也为  $T$ ，但是可以通过调节，把亮灯时间调整为  $n \cdot T$ 。

注 2：traffic\_flow\_table 中的 traffic\_flow 字段，是一个 1680 维的二位数组，从 traffic\_flow[0]，表示在  $t_0$  时刻，也即 6 点 0 分 0 秒-6 点 0 分 30 秒内，从 FromID 进入 TrafficLightID 的车流量。

红绿灯前分流情况，可以由  $\alpha$ （直行概率）， $\beta$ （左转概率）， $\gamma$ （右转概率）参数控制， $\alpha + \beta + \gamma = 1$ 。为了简化期间，假设它们是固定值，与红绿灯无关。

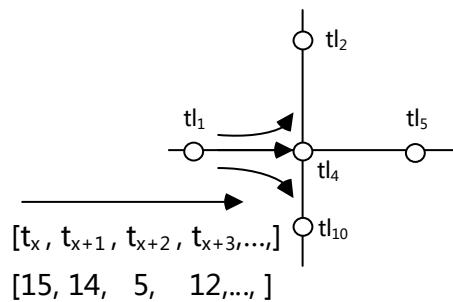


Figure 2: 红绿灯分流示意图

如图 2 所示，在  $t_x$  时刻，有 15 辆车从  $tl_1$  到达  $tl_4$ ，假定直行概率  $\alpha = 80\%$ ，会有 12 辆车需要直行，假定  $\beta = \gamma = 10\%$ ，一辆车可能左转，一辆车可能右转。

每个红绿灯口，绿灯时，单位时间内可以通过的车辆，在初赛阶段为了简化起见，设置成全局相同的值。

$\alpha$ ， $\beta$ ， $\gamma$  以及单位时间内通过的车辆数，都是参数，详见评分代码。

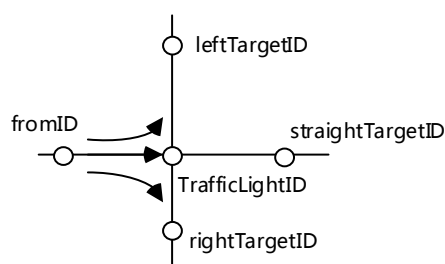
## 3.4 评分方法

我们以给定时间段内，所有等待车辆的等待单位时间总和作为 Penalty，Penalty 最小得分越高。但是有一些情况下，我们会扣分（增加 penalty 的值），这些情况包括：

### 3.4.1 违反公平性原则的扣分

每一个红绿灯，不能亮超过  $4T$  时间的红灯，也即每  $4T$  时间必须切换一次绿灯。如果违反该原则，penalty 会加上该红绿灯前等待的车辆数（ $v$ ）及连续红灯时长( $r$ )的一个函数值： $penalty += v \times \sqrt{r - 4}$

### 3.4.2 违反交通规则扣分



1. 垂直方向不能同时直行规则：fromID 到 TrafficLightID 的 straightLightStatus 为绿灯时，如果参赛者设置 leftTargetID 到 TrafficLightID 的 straightLightStatus 也为绿灯，或者设置 rightTargetID 到 TrafficLightID 的 straightLightStatus 为绿灯，则违反了垂直方向不能同时直行的规则，penalty 会加上三个方向红绿灯当前时刻车辆数之和的倍数作为扣分，该倍数是参数值，将在评分代码中公开。
2. 直行时垂直方向不可左转：fromID 到 TrafficLightID 的 straightLightStatus 为绿灯时，如果参赛者设置 rightTargetID 到 TrafficLightID 的 leftLightStatus 也为绿灯，则违反了直行时垂直方向不能左转的规则，penalty 会加上这两个方向红绿灯当前时刻车辆数之和的倍数作为扣分。

## 四. 表结构定义

### 4.1 traffic\_light\_table：红绿灯节点关系图。

表结构

字段	类型	注释
TrafficLightID	Integer	当前红绿灯编号
FromID	Integer	来源节点编号
leftTargetID	Integer	左转目标节点编号
rightTargetID	Integer	右转目标节点编号
straightTargetID	Integer	直行目标节点编号

## 4.2 Traffic\_light\_status\_table ( t<sub>x</sub> ) : t<sub>x</sub>时刻红绿灯状态表

表结构

字段	类型	注释
TrafficLightID	Integer	当前红绿灯
FromID	Integer	来源节点
leftLightStatus	Integer	左转红绿灯,1 为绿灯，0 为红灯，下同
rightLightStatus	Integer	右转红绿灯
straightLightStatus	Integer	直行红绿灯

## 4.3 traffic\_flow\_table : 车流量情况

表结构

字段	类型	注释
----	----	----



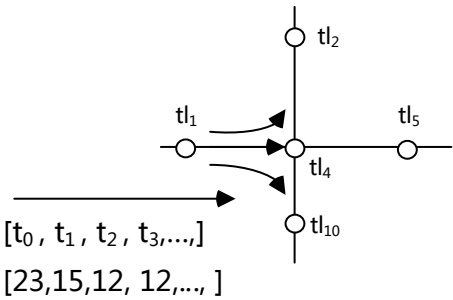
TrafficLightID	Integer	当前红绿灯
FromID	Integer	来源节点
trafficFlow	Array	在一个时间段内的车流量情况，定长。 数组长度等于时间段数。

#### 4.4 vehicle\_through\_rate\_table：单位时间红绿灯通过率

表结构

字段	类型	注释
TrafficLightID	Integer	当前红绿灯
FromID	Integer	来源节点
leftThroughRate	Integer	绿灯时左转单位时间可通过的车数量
rightThroughRate	Integer	绿灯时右转单位时间可通过的车数量
straightThroughRate	Integer	绿灯时直行单位时间可通过的车数量

### 五. Penalty 计算方法举例



假设 $\alpha = 80\%$  ,  $\beta = 10\%$  ,  $\gamma = 10\%$

假设从 tl1 方向进入 tl4 的直行通过率为 35 辆车 , 左转通过率为 2 辆车 , 右转通过率为 2 辆车。

**traffic\_light\_table :**

红绿灯	来源节点	左转目标节点	右转目标节点	直行目标节点
tl <sub>4</sub>	tl <sub>1</sub>	tl <sub>2</sub>	tl <sub>10</sub>	tl <sub>5</sub>

**traffic\_light\_status\_table ( tx ):**

**t<sub>0</sub> 时刻**

红绿灯	来源节点	左转红绿灯	右转红绿灯	直行红绿灯
tl <sub>4</sub>	tl <sub>1</sub>	0	0	1

**traffic\_flow\_table**(tl<sub>4</sub>, tl<sub>1</sub>, t<sub>0</sub>) = 23 , 也就是有 23 辆车在这个时刻到达

红绿灯 tl<sub>4</sub> , 根据直行概率 $\alpha = 80\%$  , 会有 18 辆车需要直行 , 2 辆车可能左转 , 3 辆车可能右转。根据 traffic\_light\_status\_table , 直行是绿灯 , 可全部通过 , 左转和右转是红灯 , 不可通过 , 所以

Penalty(tl<sub>4</sub>, tl<sub>1</sub>, t<sub>0</sub>) = 5。

**t<sub>1</sub> 时刻**

红绿灯	来源节点	左转红绿灯	右转红绿灯	直行红绿灯
tl <sub>4</sub>	tl <sub>1</sub>	0	1	1

**traffic\_flow\_table** (tl<sub>4</sub>, tl<sub>1</sub>, t<sub>1</sub>) = 15 , 根据直行概率 $\alpha = 80\%$  , 会有 12 辆车

需要直行，2 辆车可能左转，1 辆车可能右转。根据 traffic\_light\_status\_table，直行是绿灯，可全部通过；左转是红灯，不可通过；右转是绿灯，可以通过，但是右转此时已有 4 辆车等待，而右转通过率为 2，所以仍有两辆车无法通过。所以  $\text{Penalty}(tl_4, tl_1, t_1) = 4$ （左转区等待车辆）+ 2（右转区等待车辆）。总体  $\text{Penalty} = 5 + 6 = 11$ 。

## 六. 表结构定义

Q1. 左转概率，右转概率和直行概率为全局固定值是什么意思？

答：在初赛的时候在所有路口以上概率都是一样的，并且给定。在复赛的时候每个路口的概率可能会有所不同。详见最终复赛文档。同样的，路口的通过率，在初赛阶段，为了降低难度，也是固定的，但是复赛的时候，可能会根据实际交通观察来设定。

Q2.最后的测试流程是怎么样的？

答：测试过程为：

1. 模拟器调将当前流量作为标准输入传给参赛者，然后参赛者的程序将当前红绿灯状态作为标准输出
2. 模拟器根据当前红绿灯状态计算 penalty,并根据流量公式计算出下一时刻的各个路口的流量。
3. 跳转第一步。

Q3.流量计算公式考虑了哪些因素？

答：主要考虑了两方面的因素，一个是上一个时刻从其他路口过来的流量（根据参赛者给

出红绿灯状态模拟), 另一个是该路口实际观察到的流量。这两方面的流量会根据一定的规则融合。

Q4. 给定当前的 TrafficFlow 以及 TrafficLightStatus, 那么下一个时刻的 TrafficFlow 对参赛者而言是不是可以计算的?

答: 是的, 流量计算公式是给定( 请参照文档)。所以参赛者可以自己编写模拟器用于测试( 训练时)。但是有一点就是“测试时”每个路口的观察流量对参赛者是不可知的, 参赛者只能通过历史流量来预测。

Q5. 目前的交通规则是不是所有路口都可以右转?

答: 是的。但是必须考虑到右转带来的问题, 比如给下一个已经拥堵的路口雪上加霜。

Q6. 历史数据怎么使用?

答: 流量计算公式里面有一部分是路口的观察流量, 这些流量每天都不一样。而测试的时候这些流量对参赛者是不可见的, 参赛者只能通过历史数据来估计这部分值。同时参赛者可以根据历史数据, 分析贵阳市的道路拥堵的时间特点与空间特点, 相信可以用在最后的测试中。

Q7. 历史数据是否会和流量计算公式产生冲突?

答: 历史数据指的其实是路口观察到的流量, 它只是流量计算公式的一部分, 所以不会和公式产生冲突。

Q8. 如果红绿灯的设置，违反了交通规则怎么办？

答： 虽然在实际的生活里，不允许违反交通规则，但是为了简化比赛，我们允许违反交通规则的状态设置，但是会对违反的情况做处罚。