交叉口可变导向车道和信号控制算法

——兼顾效率优先/车辆优先通行

|  |  |  |  |
| --- | --- | --- | --- |
| 时间 | 版本 | 人员 | 备注 |
| 2021-1-7 | V0.1 | 韦学武 | 算法的设计文档 |
| 2012-1-8 | V0.2 | 智联和车联网中心 | 对接需求 |

# 项目功能需求

1. **动态车道管理**。针对交叉口的拥堵问题，通过交叉口处的动态划分车道功能可以实现对交叉口进口道的空间资源进行实时地合理分配；
2. **协作式交叉口信号控制**。从一系列微观的车辆运行和转向数据，分析路口信号灯各相位对应的交通流状况，并实时调整和优化信号控制策略；
3. **被动式公交优先信号控制**。结合路口交通流情况，计算公交车辆到达路口时间，控制信号灯，实现提前配置信号灯优先相位的绿灯加长或或其他相位红灯减少，以保证公交车辆能够高效通过路口。

# 项目数据格式

## 算法输入

表1 数据需求说明

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 字段名称 | 简写 | 数据类型 | 备注 |
| 车辆信息 | 时间戳 | sec | double |  |
| 交叉口编号 | cross\_id | int |  |
| 车辆ID或号牌 | id | int/string | 可选 |
| 车辆类型（没有） | car\_type | string | 可选 |
| 车道编号 | lane\_no | int |  |
| 车辆经度 | longtitude | double |  |
| 车辆纬度 | lattitude | double |  |
| 车辆高程 | altitude | double |  |
| 方向角 | arc | double |  |
| 速度 | speed | double |  |
| 统计信息（自己算，提供lib库） | 车道排队长度 | queue\_length | double |  |
| 车道排队车辆数 | queue\_num | int |  |
| 车道/路段速度 | road\_speed | double |  |

表2 车辆优先通行需求数据

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 简写 | 数据类型 | 备注 |
| 时刻 | time\_stamp | double |  |
| 请求车辆ID或车牌 | id | int/string |  |
| 车辆经度 | longtitude | double |  |
| 车辆纬度 | lattitude | double |  |
| 车道编号 | lane\_no | int |  |
| 车辆速度 | speed | double |  |
| 航向角 |  | double |  |
| 请求类型 | request\_type | enum（优先通行…） | 可选 |
| 请求有效时间 | request\_time | double | 可选 |
| 公交线路路径 | route | string |  |
| 请求交叉口ID | cross\_id | int |  |
| 请求信号灯相位ID | request\_phase | int |  |

表3 地图需求数据

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 字段名称 | 简写 | 数据类型 | 备注 |
| 节点 | 区域编号 | region | int |  |
| 节点ID | nodeid | int |  |
| 坐标 | refPos | struct |  |
| 路段 | 路段名称 | name | string |  |
| 上游节点\_区域编号 | region | int |  |
| 上游节点\_节点ID | nodeid | int |  |
| 路段宽度 | linkWidth | double |  |
| 路段限速\_限速类型 | speedlimitType | int |  |
| 路段限速\_限速值 | speed | double |  |
| 路段中间点 | points | struct |  |
| 路段转向\_下游节点\_区域编号 | region | int |  |
| 路段转向\_下游节点\_节点ID | nodeid | int |  |
| 路段转向\_相位编号 | phaseId | int | SPAT提供 |
| 车道 | 车道编号 | laneId | int |  |
| 车道宽度 | laneWidth | double |  |
| 车道属性\_车道类型 | laneType | int |  |
| 车道转向 | maneuvers | struct |  |
| 下游车道\_下游节点\_区域编号 | region | int | 可选 |
| 下游车道\_下游节点\_节点ID | nodeid | int | 可选 |
| 下游车道\_下游车道编号 | laneId | int | 可选 |
| 下游车道\_驶入下游车道的转向 | maneuvers | struct | 可选 |
| 下游车道\_当前车道的相位编号 | phaseId | int |  |
| 车道限速\_限速类型 | speedlimitType | int |  |
| 车道限速\_限速值 | speed | double |  |
| 车道中间点 | points | struct |  |

表4 信号控制方案数据

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 字段名称 | 简写 | 数据类型 | 备注 |
| 信号机状态 | 现有控制方案编号ID | intersectionId | int |  |
| 信号机的工作状态指示 | status | int |  |
| 相位状态 | 相位编号 | id |  |  |
| 相位信号灯状态 | light | int |  |
| 相位下一次开始的时间 | startTime | int |  |
| 相位当前状态或下一当前相位的最短结束时长 | minEndTime | int |  |
| 相位当前状态或下一当前相位的最大结束时长 | maxEndTime | int |  |
| 当前状态的准确结束时间或预测结束时间 | likelyEndTime | int |  |

## 算法输出

表5 路侧引导和控制数据

|  |  |  |  |
| --- | --- | --- | --- |
|  | 字段名称 | 简写 | 数据类型 |
| 可变导向车道 | 车道控制方案编号ID | controller\_id | int |
| 车道功能 | movement | enum |
| 车道控制开始时间 | lane\_start\_time | datetime |
| 车道控制结束时间 | lane\_end\_time | datetime |
| 协作式通行 | 控制方案编号ID | controller\_id | int |
| 控制方案开始时间 | controller\_start\_time | double |
| 控制方案结束时间 | controller\_end\_time | double |
| 周期时长 | cycle | int |
| 相序信息 | phase\_order | string |
| 相位绿灯时长 | green\_time | int |
| 相位红灯时长 | yellow\_time | int |
| 相位全红时长 | all\_red\_time | int |

# 算法设计

## 可变导向切换

通过判断直行和左转是否处于拥堵状态，以及比较左转和直行的拥堵程度决定是否切换车道功能。利用车道实际通行流量与车道通行能力的比，即饱和度，作为拥堵状态判断的依据，并将阈值设为0.8（根据不同的交叉口情况予以调整）。



式中：

——观测时间T内流量值，单位为标准小汽车量/观测间隔；

### 判断指标

流量是指观测时间内，通过道路指定断面或车道的车辆数。

流量的统计周期设置为5分钟，具体计算公式参照如下：



式中：

——观测时间T内流量值，单位为标准小汽车量/观测间隔；

——采样间隔内标准小汽车量，单位为标准小汽车量/采样秒；

——车辆的类型总数；

——车辆类型对应的标准小汽车换算系数，参见附录6.1；

——采样间隔内某一车辆类型的车辆数，单位为（辆/采样秒）。

通行能力是指道路与交通处于理想情况下，每一条车道（或每一条道路）在单位时间内能够通过的最大交通量。

通行能力的统计周期设置为1小时（保证采集充足的车头时距数据），具体计算参考如下：





式中：

——某一时段（通常一小时），考虑信号控制的影响，车道的基本通行能力值，单位为标准小汽车量/小时；

——某一时段（通常一小时），车道的基本通行能力值，单位为标准小汽车量/小时；

、、、——分别是相位的绿灯时长、黄灯时长、全红时长，以及信号控制周期时长，单位：秒；

——某一时段，车道的饱和车头时距，单位为秒；

——车辆类型m的车辆数；

——同一车道上，车头时距小于等于4的前后两辆车的车头时距值，单位为秒；

——车辆类型m的标准小汽车换算系数；

### 切换逻辑

左转和直行切换的核心是在满足最小切换周期的条件下，判断左转和直行的饱和度是否满足切换阈值和。

1. 定义切换周期为流量统计时长，选取切换周期N作为可变导向车道被触发的阈值。当直行变左转时取N≥6，当左转变直行时取N≥4；
2. 饱和度的比较。假设车道的初始属性为直行，若连续N个周期内左转饱和度≥，直行饱和度<，且切换后左转直行饱和度的差值小于，则触发可变导向车道由直行切换为左转；反之，若直行饱和度≥，左转饱和度<，且切换后左转直行饱和度的差值小于，则触发可变导向车道由左转切换为直行。

### 过渡状态

过渡状态的核心是车道的排队车辆进行清空操作，同时禁止上游车辆持续驶入。

1. 当车道功能分配方案需要变更，则将具体变更功能的车道切换为过渡状态，禁止车辆驶入；
2. 确认变更功能车道已清空后或者当前周期执行完毕，则将具体变更功能的车道切换为最终变更状态。当无法获取排队长度信息时，按照渠化段的最大长度计算可排队车辆数，计算所需的绿灯清空时间。

## 交叉口信号控制算法

### 算法框架

交叉口的控制算法采用分布式控制思路，基于决策树的方法，选择最佳的相序方案。寻找可行的相序相位方案的过程，就是建立一个决策树的过程。决策树的每个节点表示相序的某一阶段（由两个相位组成，参见附录6.2），决策树的分支表示延续或者跳转相位。当决策树到达叶节点时，形成一条可行的相序方案。根据预测的流量和车辆延误，通过深度优先搜索和回溯的方法选择最佳的相序以及相位绿灯方案。

每个交叉口需要提前获取车道的流量预测值或车辆到达分布、排队长度值，并且为实现车辆优先，每辆车具有不同的权重，决定着车辆清空时间和延误的计算，从而达到车辆被动优先通行（参见附录6.3）的目的。

具体的算法步骤如下：

1. 相序嵌套矩阵。根据交通流向的冲突关系，通过相位嵌套或搭接矩阵表示相序方案的可行空间，建立决策树的树形路径，具体优化的算法后文描述；
2. 相位绿灯初始化。已知决策树某一相序次序顺序，每个交叉口合理的周期范围和每个相位的清空比例上下限，进行二分搜索周期区间。对于每个可行的初始周期值，根据两个嵌套相位的预测流量值或者上游车辆达到分布（考虑排队车辆的影响）得到当前阶段的交通需求，同时保证相位绿灯下车辆清空比例的约束（每个相位的清空比例可调），计算相位绿灯初始值；
3. 绿灯时长的计算。基于相位绿灯初始值和最小绿灯时长，确定相位绿灯的可行区间；对于每个可行值，预估当前相位和后续相位车辆在停止线前的累积延误时间，计算车辆的总延误，从而基于贪心算法寻找最佳的绿灯时长方案；并更新相序路径的延误值；
4. 周期时长的调整。当完整相序的绿灯总时长超过或低于初始周期时，通过初始周期与总时长的比例，调整相位的清空比例（优先通行相位的同步提高或保持1不变），并计算当前相序路径下的总延误；
5. 优化算法。分两个阶段进行，来选择最佳的相序相位方案。一阶段为深度搜索寻找初始路径，二阶段为回溯过程。



图1 算法实现的流程

### 相序嵌套矩阵

相序优化的基本框架主要有两种切换的方法：基于信号灯组以及基于信号状态。基于信号状态（即单环结构）的相序方案，无法处理交叉口对向不均衡交通流的优化场景，单独优化每个相位的绿灯时长存在实现困难。因而采用NEMA标准的基于信号灯组方案，采用非对称双环结构相位对交叉口进行信号相序的优化。以十字交叉口为例，如下图所示：

图2a 十字交叉口的八相位方案 图2b 某一相序组合

按照图2a所示，八相位方案组成的相序可排列组合成16种不同的相序组合方案，图2b为其中三种组合方案。

参照附录6.4的相位搭接和嵌套规则，列举各相位的嵌套关系，形成相应的嵌套矩阵（右转与直行嵌套放行）：

表1 可嵌套的双环结构

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 相位 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 |  | ▲ |  |  | ▲ |  |  |  |
| 2 | ▲ |  |  |  |  | ▲ |  |  |
| 3 |  |  |  | ▲ |  |  | ▲ |  |
| 4 |  |  | ▲ |  |  |  |  | ▲ |
| 5 | ▲ |  |  |  |  | ▲ |  |  |
| 6 |  | ▲ |  |  | ▲ |  |  |  |
| 7 |  |  | ▲ |  |  |  |  | ▲ |
| 8 |  |  |  | ▲ |  |  | ▲ |  |

采用邻接表存储上述的嵌套结构，即通过数组与链表相结合的方式存储：

1. 相位用一个一维数组存储。存储相位的信息和指向下一个嵌套相位的指针；
2. 每个相位的嵌套相位构成一个线性表，采用单链表存储。

### 相位时长初始化

每个交叉口设置合理的周期时长范围，可以通过设置上界和下界指定周期的取值范围（比如公共周期），可以按照如下的规范执行：

1. 信号周期时长在非饱和交通状态下，宜设置在 30秒-150秒范围；
2. 在饱和交通状态下，不宜超过180秒。

以延误为评价指标时，信号周期时长存在最优解（如下图所示），因而按照二分查找算法，选择周期时长的值，可以实现循环查找最佳的周期值。

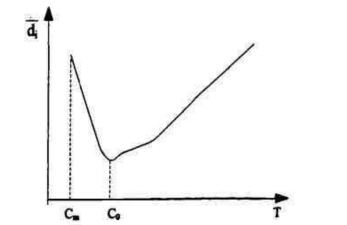


图3 信号周期与延误的关系

将预测流量值或者上游车辆到达分布、初始排队车辆数和和每个相位的清空比例上下限作为主要依据，分配嵌套相位的初始绿灯时长。按照如下的计算公式，确定嵌套相位的初始候选绿灯时长：



式中：

——相位i的初始候选绿灯时长，单位：秒；

，——嵌套相位下基于清空比例约束每个相位的车道流量值的最大值，以5分钟为统计间隔，单位：pcu/5min；

——统计间隔，单位：秒，以5分钟为准；

——周期时长，单位：秒；

——相位i的饱和车头时距值，单位：秒，采用安全车头时距确定默认值，或者基于历史数据的统计值；

——相位i在时间间隔开始时的排队车辆数（基于清空比例调整），单位：pch；

——嵌套相位下对称交通流量的阈值，默认为0.1；

考虑到每个相位i启动阶段，存在连续若干辆车的启动延误时间，计算公式为：



以及如下相位时长的约束条件：

1）街道及交叉口的几何条件下安全过街最小绿灯时长。



式中：

——人行横道长度，单位：米；

——行人过街步速，单位：米/秒，设计步速为1.0m/s；

——相位i的绿灯间隔时间，单位：秒。

因而初始绿灯的修改值为



### 周期时长调整

交叉口的每个相位按照初始绿灯分配时长，存在完整相序的绿灯总时长超过或低于初始周期时，通过初始周期与总时长的比例，调整相位的清空比例（优先通行相位的同步提高或保持1不变），并计算当前相序路径下的总延误。

### 延误计算

决策树算法的目标函数是最小化所有车辆的累积延误值。在计算延误过程汇总，考虑不同车辆类型的延误权重，便于实施公交优先或特种车辆优先通行。

对于交叉口红灯、黄灯、绿灯的状态，分不同情况计算每辆车在交叉口的延误值。具体情况主要有三类延误：因红灯停车等待的延误、绿灯开始前方排队车辆启动延误、绿灯开始前方排队车辆清空延误等，不考虑车辆随机到达产生的延误，具体计算公式如下：







式中：

——当前所处相位的编号；

——从起始相位到当前相位之间的相位编号；

——不同类型的标准小汽车，单位：pcu；

——不同类型的标准小汽车对应的延误权重；

——车辆是否处于相位i的绿灯状态，1表示是，0表示否；

——默认的启动延误值，单位：秒；

——相位i的饱和流率，单位：pch/h；

——相位i的绿灯时长，单位：秒；

——周期长度，单位：秒。

因而每个相位i的总延误表示如下：



在每个相位结束时刻，可根据车辆所处的情况，分别计算交叉口所有车辆的延误值。从算法优化的角度，延误计算包含两部分，一个是从根结点到当前节点，所有车辆的累计延误和，另一个是从当前节点到叶子节点的累计延误值。前者用于评估已选方案，后者用于评估最优方案。

### 相位时长优化

基于相位绿灯初始值和最小绿灯时长，确定相位i绿灯的可行区间。

假设为满足一辆车的通行，所有相位的单位时间步长设为。为保持周期长度不变，当前相位下每缩短，则其他后续相位按剩余清空车辆的优先级分别增加相位绿灯时长（有车辆优先通行标记的相位，只增加相位绿灯时长）。进而，按照延误计算的方法，分别计算当前相位和后续相位车辆在停止线前的累积延误时间，计算车辆的总延误。

在相位i绿灯的可行区间按遍历过程中，基于贪心算法寻找最佳的绿灯时长方案，如果延误值小于当前的累计延误，则更新相序路径的延误值，继续遍历；否则停止遍历，循环下一相位i+1，当循环至优先通行相位时跳过。

### 搜索算法

在建立决策树的过程中，根据累计延误搜索最优的相序方案。当到达决策树的某一内部节点时，从一条分支出发，深度搜索直至叶节点，当前路径的搜索停止（如下图所示）；并回溯搜索下一分支的路径。递归计算累计延误的公式为：



式中：

——相位阶段s结束时，所有车辆的累计延误值，单位：秒；

——相位阶段s+1下，相位i的车辆延误值，单位：秒；



图 相序的决策树示例

决策树采用二叉树作为算法的基础，左分支表示保持当前ring1的相位，右分支表示切换当前ring1相位的下一相位，如下图所示。其中在当前相序阶段s下，左分支存在的充分条件是



同时，记每一相位阶段s绿灯开始时，输入状态由一组向量组成，如果相位i处于绿灯状态，驶出的排队车辆和到达车辆为





则当进入左分支或右分支的相位阶段s+1时，更新向量：



式中：

——系统的绝对时间值。

为避免相同相序阶段下延误的重复计算，建立延误数组，用于存储指定流量下相序阶段结束时刻的延误值。



虽然通过回溯能实现全局最优的搜索，但计算量较大。为了减少回溯穷举的搜索计算量，采用剪枝的操作，当左分支或右分支的相序阶段最大流量为0时满足剪枝条件，



即不再深度搜索分支，从而实现快速地寻找最优的相序相位方案。

# 时间安排（2人）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 工作内容 | 工作推进思路 | 输出 | 开始时间 | 结束时间 |
| 1 | 动态车道管理 | 算法调研 | 算法设计文档 | 2020-12-16 | 2020-12-25 |
| 指标计算方案 | 指标计算文档（包含计算方法、阈值标定） |
| 代码开发 | 代码 | 2021-1-11 | 2021-1-20 |
| 本地调试和优化V1.0 | 代码、数据库 |
|  |  | 算法仿真验证、文档整理 | 算法交付文档（包含功能、函数说明、调用说明、仿真测试结果） | 2021-1-25 | 2021-2-5 |
| 2 | 协作式交叉口信号控制 | 算法调研 | 算法设计文档 | 2020-12-25 | 2021-1-8 |
| 算法可行方案 | 算法开发文档（包含数据、技术方案、功能说明） | 2021-1-11 | 2021-1-29 |
| 完成30%代码开发 | 代码 | 2021-1-25 | 2021-2-9 |
| 完成70%代码开发 | 代码 | 2021-2-22 | 2021-3-26 |
| 本地调试和优化V1.0版 | 代码、数据库 | 2021-3-29 | 2021-4-9 |
| 算法仿真验证、文档整理 | 算法交付文档（包含功能、函数说明、调用说明、仿真测试结果） | 2021-4-12 | 2021-4-23 |

# 参考文献

【孙强，动态可变车道优化算法研究】，采用流量作为可变车道算法的参数，并标定路段行程时间的阻抗，利用最优化算法分配最优的车道组合。计算交通流的方向分布系数大于2/3，是设置可变车道的基本依据。

【周鹏、丁晨，智能可变车道的导向判决算法的研究与实现】采用流量与通行能力作为判断依据，以流量/通行能力大于90%作为左转/直行的阈值，并比较左转和直行的拥堵程度选择变换的车道。

【刘昱岗、王卓君等，基于ANFIS的可变导向车道智能控制系统】利用线圈检测器，将交通量和占有率作为输入，采用模糊神经网络ANFIS进行直行/左转状态（4个状态）的预测；判断的周期为2-3个信号周期长。

【蔡云、杨晓光等，一种灵活的在线交通信号相位切换结构】建立相位搭接和嵌套的6个规则，基于NEMA TS2的相位定义方法，重点穷举了8相位情况下的相位切换空间（16个相位顺序\*64个切换路径）。

【聂磊，马万经等，信号控制交叉口相位相序自动生成和优化模型】采用两阶段方法，依次优化关键车流的相位、非关键车流的相序。重点通过定义5个参数：车道冲突矩阵、车流冲突矩阵、相位矩阵、相序排列集合、绿灯间隔时间矩阵，建立相位相序的可行解空间。并建立自定义的约束不等式，优化最小绿灯间隔时间。

【肖人彬、王颖聪，一种面向时间分配问题的群智能劳动分工新方法】将信号配时问题作为时间分配任务，并将车均延误、停车次数分别对应为蜂群模型的激发剂-抑制剂概念，通过相位时间的增加、减少、保持动态调整配时。

【Xiao-Feng Xie、Stephen F.Smith等，Schedule-driven intersection control，2012】

【Xiao-Feng Xie、Stephen F.Smint等，Schedule\_driven coordination for real-time traffic network control，2012】

【I.Porche、S.Lafortune等，Adaptive Look-ahead Optimization of Traffic Signals，1997】

【Biao Yin等，Forward search algorithm based on dynamic programming for real-time adaptive traffic signal control，2015】

# 附录

## 车辆类型换算

参考JTG B01-2014 《公路工程技术标准》

表 车辆类型对应的标准小汽车换算系数

|  |  |  |  |
| --- | --- | --- | --- |
| 车型 | 流量换算系数 | 公交优先下延误换算系数 | 说明 |
| 小客车 | 1.0 | 1.0 | 座位≤19座的客车和载重量≤2t的货车，**车长≤6m** |
| 中型车 | 1.5 | 1.0 | 座位＞19座的客车和2t＜载重量≤7t的货车，**6m＜车长≤9m** |
| 大型车 | 2.5 | 1.0 | 7t＜载重量≤20t的货车，**9m＜车长≤12m** |
| 公交车 | 2.5 | 2.5 | **9m＜车长≤12m的**公交车 |
| 汽车列车 | 4.0 | 1.0 | 载重量＞20t的货车，**车长＞12m** |

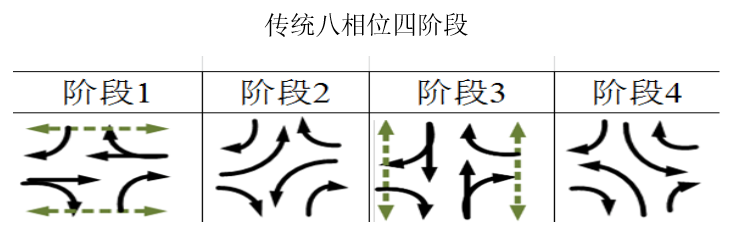
## 阶段和相位

相位阶段（Stage）：

根据交叉口通行权在同一周期内的更迭次数来划分，一个信号周期内通行权的交接几次，就是几个信号阶段。

相位（Phase）：

按照车流获得信号显示的时序来划分。在现有的信号机中，对应每一股车流都会有一个相位进行控制。



## 主动优先和被动优先

主动优先就是由公交车主动触发优先信号，当公交车即将到达路口时给信号机发送指令，保障公交车到达路口时信号灯为绿灯，这种模式信号灯变化随机，对社会车辆通行影响较大；

被动优先就是通过信号灯定时给出优先放行信号，公交车被动接受信号优先，此模式信号配时相对固定，对社会车辆通行影响相对较小。

由于主动优先对社会车辆影响较大，公交信号灯优先控制多采用被动优先控制模式，对于被动优先控制采取了两种不同的控制策略：

1. 公交信号灯提前绿灯控制。通过路口设置的公交信号灯控制公交车提前若干秒绿灯通过路口，此控制模式对社会车辆的影响就是压缩了若干秒通行时间；
2. 公交绿波带控制。对于公交站台建设在路口上游的路口，因为站台上下客时间不可控，无法考虑绿波控制，此控制模式对社会车辆的影响相对较小。

## 相位搭接和嵌套规则

NEMA TS2 对相位搭接（overlap）的定义为一股或一股以上交通流在绿灯以及清空时间内跨越了两个或两个以上相位。 为构建一个结构清晰、各股交通流之间互不冲突的相位切换方案，在加入行人、非机动车相位以及右转专用相位的情况下，制定基于双环结构的相位搭接及嵌套规则：

1. 为了避免第一类冲突点（非共用信号车流之间的冲突点），不同相位组之间的左转及直行车流不能形成搭接相位。
2. 为了避免第二类冲突点（共用信号转弯车流之间的冲突点），左转相位不能与统一相位组的对向直行车流行程搭接。
3. 行人与非机动车相位与紧邻的非冲突直行相位形成嵌套相位。
4. 右转专用相位嵌套规则。在下述两种情况应考虑设置右转专用相位：
5. 相交道路出口道不足，右转车流与其他车流合流放行时造成冲突；
6. 行人与非机动车过街需求比较大，右转车流不受信号灯控制时转弯

速度较快，容易与行人和非机动车形成冲突。

右转专用相位与其他相位的嵌套规则可概括为两条：

1. 右转专用相位与相交道路的邻近左转相位形成固定嵌套相位；
2. 右转专用相位与毗邻直行相位形成感应嵌套相位，与之冲突的行人相

位需求不大，且其本身也有一定需求时可以考虑设置。