**基于蒙特卡洛模拟优化的演唱会交通网络评估**

**摘要**

今年汕头多次开展演唱会，在带来良好的经济文化效应的同时，造成了一个不容忽视的问题——交通堵塞。针对此乱象，本小组运用蒙特卡洛对单向单车道的理想情况进行了分析，争取在解决交通拥堵问题、合理道路建设上有所贡献。

**关键词**：蒙特卡洛；蒙特卡洛树搜索；蒙特卡洛模拟；蒙特卡洛算法优化；流程图；交通建设；道路设计；行车路线预测

目录

[第一章 绪论 3](#_Toc153137484)

[1.1 研究背景 3](#_Toc153137485)

[1.2 研究目的及意义 3](#_Toc153137486)

[1.2.1 研究目的 3](#_Toc153137487)

[1.2.2 研究意义 3](#_Toc153137488)

[1.2 研究的基本思路及研究过程 4](#_Toc153137489)

[1.3.1 研究的基本思路 4](#_Toc153137490)

[1.3.2 研究过程 4](#_Toc153137491)

[第二章 对道路的基本分析和简单变换 5](#_Toc153137492)

[2.1 基础的地图元件和一些定义 5](#_Toc153137493)

[2.1.1 基础的地图元件 5](#_Toc153137494)

[2.1.2 一些简单的定义 5](#_Toc153137495)

[2.2 对于堵车的判定 6](#_Toc153137496)

[2.2.1 最简单的地图 6](#_Toc153137497)

[2.2.2 堵车、塞度及塞值的定义 6](#_Toc153137498)

[2.2.3 对最简单地图的一些研究 7](#_Toc153137499)

[2.3 岔路 8](#_Toc153137500)

[2.3.1 单项单车道岔路汽车算法 9](#_Toc153137501)

[2.3.2 单向单道岔路车辆算法的灵活运用 10](#_Toc153137502)

[2.4 堵车时车辆的换道 11](#_Toc153137503)

[2.4.1 的削减 11](#_Toc153137504)

[2.4.2 堵车换道系统 11](#_Toc153137505)

[2.4.3 岔路系统 12](#_Toc153137506)

[2.5 道路的简单变换 14](#_Toc153137507)

[2.5.1 道路伸缩 14](#_Toc153137508)

[2.5.2 道路拉直 14](#_Toc153137509)

[2.5.3 同终点问题 15](#_Toc153137510)

[2.5.4 同路问题 16](#_Toc153137511)

[第三章 车辆路线预测和道路分析 17](#_Toc153137512)

[3.1 路线预测 17](#_Toc153137513)

[3.2 蒙特卡洛算法优化环境相似性分析 18](#_Toc153137514)

[3.3 塞度预测与道路设计建议 19](#_Toc153137515)

[第四章 结论 20](#_Toc153137516)

[附：参考文献： 21](#_Toc153137517)

# 第一章 绪论

## 研究背景

为了树立城市文旅招牌的窗口，提升城市文化的影响力，带来经济活力，今年以来，汕头演唱会强势崛起，吸引了多位明星歌手来汕“打卡”。在漫天星闪，歌声嘹亮的同时，也造成了一个严重的问题——交通堵塞。

近年来，随着汕头经济的不断发展，交通堵塞的问题逐渐变得严重，在2022年，汕头交通拥堵的排行榜又有上升。交通的顺畅，直接关系到了一个城市的运行效率和经济发展速度。如果把城市比作一个人，那么良好的交通就是富有跳动旋律的心脏。

考虑到交通堵塞给人员流动、交通秩序带来了严重的影响，我们研究小组对此展开了研究。鉴于蒙特卡洛的实际用途，我们借助蒙特卡洛对演唱会兴盛所造成的交通拥堵问题进行了相关研究。

## 1.2 研究目的及意义

### 1.2.1 研究目的

演唱会正举行得如火如荼，但结束后出现了种种乱象，给汕头城市形象带来了负面影响。例如场面混乱不堪、停车秩序混乱、没有准时公交上车指引等问题。城市公共交通是城市活动的核心环节和重要载体，而汕头现发生了种种乱象。针对如此，我们研究小组在车辆流通、交通堵塞、道路建设方面运用蒙特卡洛进行研究，以便能够更加清晰地认识明白车辆的走向可能造成的堵塞问题，从而有效缓解交通堵塞的问题。

### 1.2.2 研究意义

#### 理论意义

城市交通经常会出现堵塞的问题，而本研究对于缓解交通拥堵、提高道路建设的合理性提供了一种理论上的方法——即运用蒙特卡洛进行预测，从而优化道路建设。对于运用蒙特卡洛对交通堵塞问题的分析研究具有一定的创新与贡献，优化应用了蒙特卡洛。

#### 现实意义

通过研究，一方面有利于促进城市交通流通、提高城市的出行效率、推动整个城市的运转，从而促进城市经济发展，给汕头经济文化带来更为强大的生命活力。另一方面，本论文中的研究，可以预测道路塞度，有利于人们选择不同且适宜的出行时间，从而车辆错峰，一定程度上缓解了交通拥堵，也满足了人们的出行需求。1.3 研究的基本思路以及研究过程

## 研究的基本思路及研究过程

### 1.3.1 研究的基本思路

将道路进行简单变换后，将其转化为流程图的形式，再将这个流程图转化为代码，就能用计算机进行蒙特卡洛模拟，帮助我们计算出哪一条道路更容易堵车，从而对道路建设给出建议。

### 1.3.2 研究过程

在确定课题后，我们简单想了研究思路，随后迅速展开研究。解决各种问题，我们成功地做出第一张流程图，同时进行对蒙特卡洛的优化。而后我们对岔路进行分析。攻克了各种难题，我们将岔路的车辆流通转化为流程图，优化了蒙特卡洛，并对塞度进行预测。

# 对道路的基本分析和简单变换

本章主要对单向单道的理想状态下道路上的车辆行驶进行简单的分析，我们会用流程图表示出车辆和道路的情况，本章会对道路进行简单的变换以便对道路车辆的行驶进行分析。

## 2.1 基础的地图元件和一些定义

为了方便对道路上的车辆行驶进行分析，我们简单的用示意图来表示地图。例如：

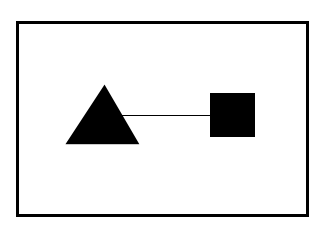


图1 最简单地图

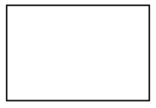
就是一个示意图。

### 2.1.1 基础的地图元件

1、“三角形”表示的是一个居民区，一般作为车辆的目的地。

2、“”表示公路，车辆在公路上行驶。

3、“正方形”表示的是演唱会的地点。

4、“”表示我们分析（考虑）的范围。

以上四个就是基础的地图元件。

基础的地图元件组成的示意图我们把它叫做地图。

（本节后不再对基础的地图元件进行介绍。）

### 2.1.2 一些简单的定义

由于在实际道路上的行驶，可能会有很多影响车辆行驶的因素，为了方便进行研究，本文对车辆的一些定义如下：

1. 对于公路：
2. 公路的长度是不确定的，在制作流程图时需要根据具体情况修改。我们用表示公路的长度。
3. 对于一条岔路，我们会用表示岔路总数，我们会对每条岔路进行编号，该条岔路的编号用表示。
4. 对于公路上的车辆，我们用表示在编号为m的岔路上车辆的数量。把在m路上的车辆进行编号，该车辆的编号用表示。用表示车辆在m路上行驶的距离。用表示驶入m路得车辆数，用表示驶出m路的数量。
5. 对于车辆：

（1）车辆速度恒定且为**10m/s**，用表示。

（2）每分钟会从演唱会开出辆车（**10≤x≤20**）。

（3）车辆的长度为。

（4）车辆在路上行驶的距离为。

## 2.2 对于堵车的判定

### 2.2.1 最简单的地图

在上一节中，我们介绍了我们基础的地图元件，仅由这四种的各一个组成的地图成为最简单的地图，它的示意图为：

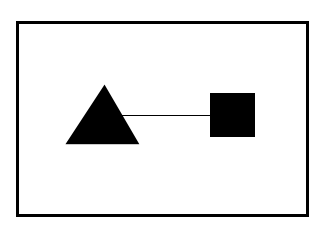


图2 最简单地图

接下来我们对这个地图进行研究并在接下来的几节进行拓展。

### 2.2.2 堵车、塞度及塞值的定义

为了方便研究，我们用**塞度**来表示一条道路的堵车程度，用希腊字母**Δ（delta）**表示塞度，其单位为1%。其定义式为

；

当**Δ≥60%**，此道路判定为堵车，堵车期间车辆不会通行，后来的车辆会选择另外的道路行驶。且此时记**塞值▽=1**。若Δ＜60%时记**▽=0**。

我们规定30min后堵车现象会消失，此时车辆可以通行。

### 2.2.3 对最简单地图的一些研究

我们先对最简单地图进行分析。在

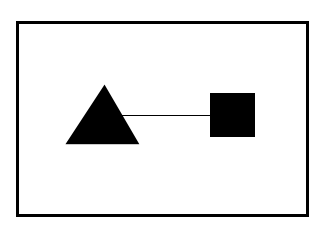


图3 最简单地图

中，车辆从“正方形”驶出，且由规定，每分钟驶出的车辆数，（表示第分钟从演唱会中驶出的车辆数）。

每过一分钟，累加1，那么我们设定初始时间为。然后令累加1，即

………………………………………（1）

此时且有

辆车从演唱会驶出，那么此时进入1路的车辆数为

驶出1路的车辆数为

于是在1路上行驶的车辆数为

那么我们对在m路上的车辆进行编号，记为

那么我们先令然后再累加1，即

………………………………………（2）

此时经过一分钟，则m路上的车形式的距离为

然后对进行判定，若，则不进行任何操作并进入下一个操作。若

则令

此判定结束后我们返回（2）步骤，直到，即所有车辆都进行了上述操作后，进入下一步。此时在m路上行驶的车辆数为

此时我们对道路的塞度进行判定，若

**

则该道路判定为不堵车，则我们记

若

**

则该道路判定为不堵车，则我们记

以上左右操作描述了第1分钟车辆行驶的情况，则我们返回到（1）重复进行以上操作，我们就能得出每分钟车辆行驶的状况。

我们把上述所有操作画成一个流程图，如下：

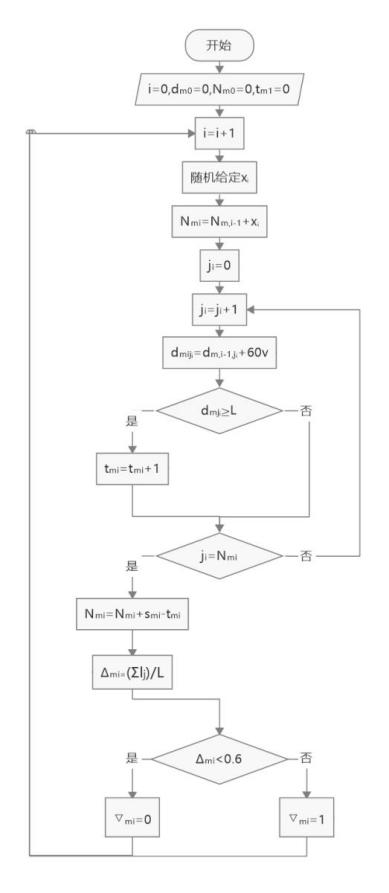


图4 单项单车道堵车判定方法（流程图）

这个流程图我们叫做**单项单车道堵车判定方法**，简单记作。

## 2.3 岔路

我们将

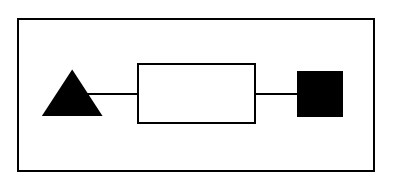


图5 简单岔路

称为简单岔路，由图可看出，此时。

### 2.3.1 单项单车道岔路汽车算法

我们把时间固定，即在进行接下来的操作时，不变。我们用表示这辆车选择的是哪条岔路，比如：若时，这辆车选择的是2号岔路。

那么我们还是先令，再累加1，即

……………………………………（1）

然后再随机取一个。那么当时，车选择m路，那么我们令，再累加1，即

……………………………………（2）

然后进行判定，若，则我们返回到（1）并重新判定

若，则驶入m路的车辆累加1，即

……………………………………（3）

直到所有车辆都选择道路，即当

后，就可以进行下一步操作（详细见下一个目）。（若，则返回到（1）步骤并继续向下运行）

我们把上述所有操作画成一个流程图，如下：

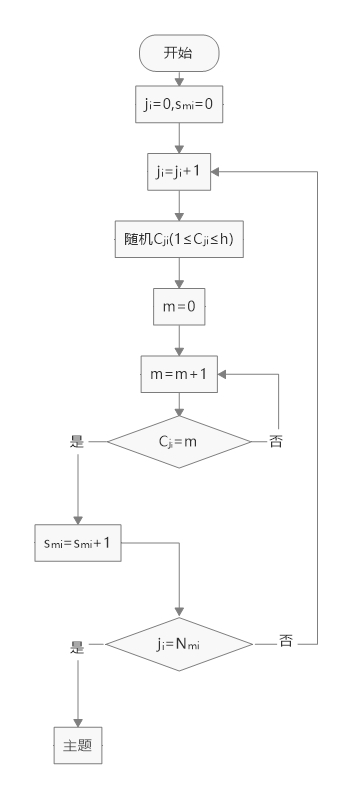
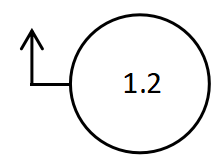


图6 单项单车道岔路车辆算法

这个流程图我们叫做**单项单车道岔路车辆算法**，记作。

### 2.3.2 单向单道岔路车辆算法的灵活运用

我们将第二节中的堵车判定方法和单向单道岔路车辆算法有机结合，结合的过程不能生搬硬套，而是要考虑各个操作的意义并进行有机结合。

我们这么结合，因为给出了岔路时车辆的先择情况，而则对一条岔路进行堵车判定，那么我们只需要在1.2之前设置变量m，并令m累加，并不断重复，直至m=h时进入下一分钟。

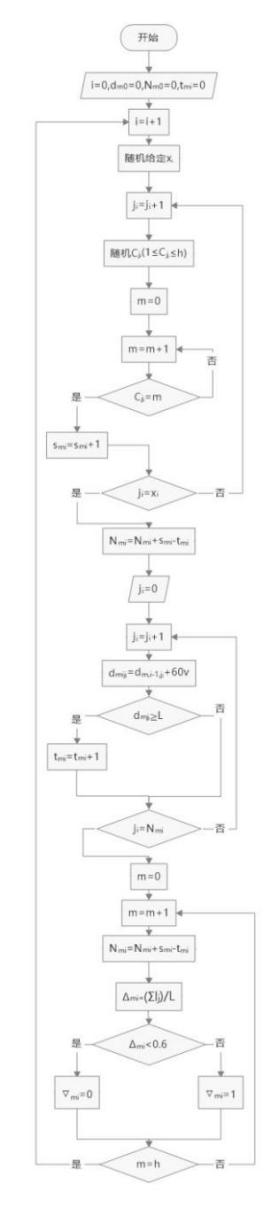
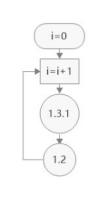


图7 单向单道岔路车辆算法的灵活运用

这个流程图简单记为。

## 2.4 堵车时车辆的换道

### 2.4.1 的削减

我们规定当时，每30min削减为0，那么我们规定每1分钟削减。

我们在判定堵车之前先对进行判定，若

，

则进行堵车判定，若

，

即该道路还处于堵车状态，则我们不进行堵车判定并令

，

该判定后进入下一分钟，将上述操作化成流程图，如下：

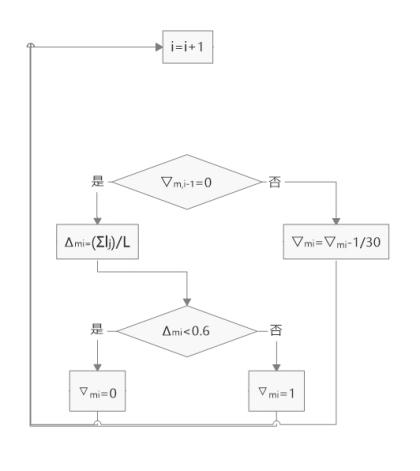
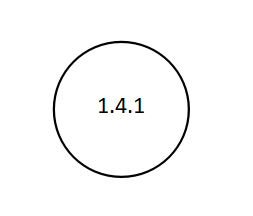


图8 塞值削减办法

这个流程图我们叫做**塞值削减办法**，简单记为。

### 2.4.2 堵车换道系统

当m路有车时，有定义我们有，于是在单项单车道岔路车辆算法中，我们在车辆选择岔路后（还未对驶入m路的车辆累加之前），再次进行判定，若



则返回到（2）步骤，然后继续向下运行，若



则我们进行下一步操作，也就是进行（3）步骤，然后继续向下运行。

那么我们给出如下的流程图：

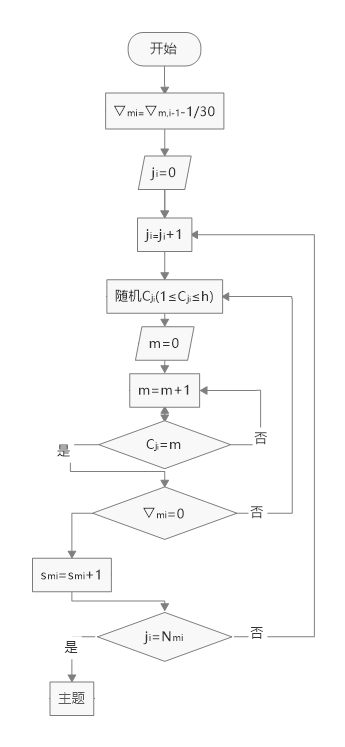
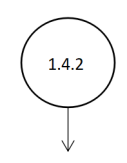
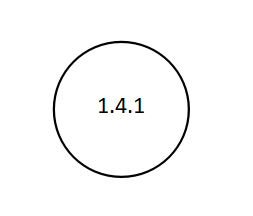
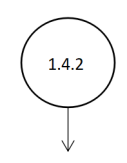
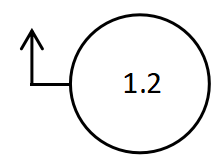
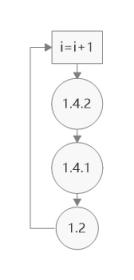


图9 岔路堵车换路系统

我们把这个流程图叫做。**岔路堵车换路系统**简单记为。

### 2.4.3 岔路系统

于是我们将、与结合起来，简单记为：

或。

其具体流程图如下：

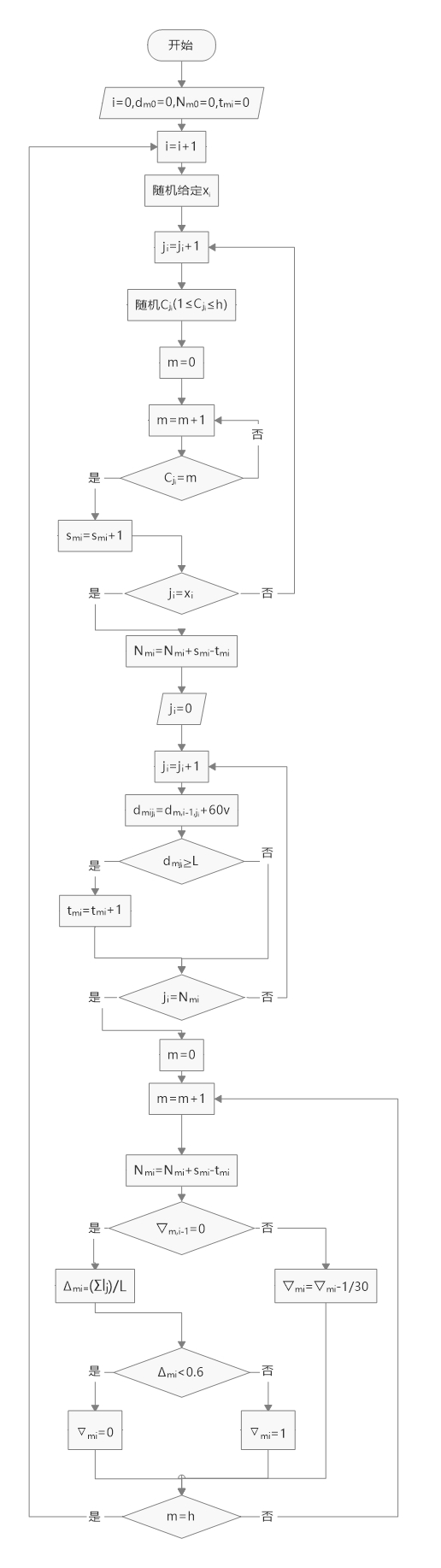


图10 岔路系统

这个流程图我们叫做**岔路系统**，它考虑了岔路时车辆的选择、堵车的判定、塞值的削减，可以基本复原岔路车辆的流通状况。而想要快速得到岔路的车流情况，还需要将流程图转化为代码，借助计算机处理这些数据。由于作者学艺不精，暂时不能实现这个操作。

## 2.5 道路的简单变换

### 2.5.1 道路伸缩

简单岔路中，我们在干路上进行伸长或缩短，对整个公路基本没有影响。于是，我们可以将影响较小的干路缩短甚至去掉，比如：

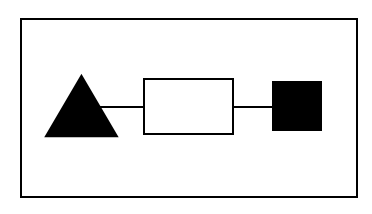


图11 简单岔路

可以变换为

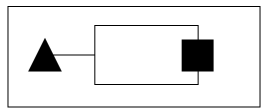


图12 简单岔路的变换

### 2.5.2 道路拉直

为了方便研究，我们暂时不考虑转弯时车辆的减速，我们可以把知道拉直，如：

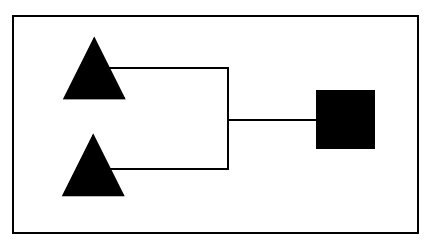


图13 例图（道路拉直）

把弯道拉直，得到

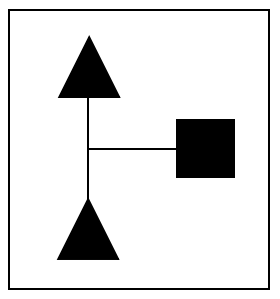


图14 例图（道路拉直）的第一次变换

最后对干路进行伸缩，变换为

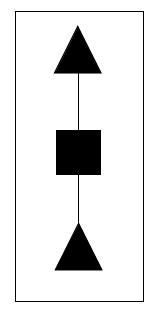


图15 例图（道路拉直）的第二次变换

### 2.5.3 同终点问题

若有两条路桐乡同个重点，则我们可以将这个终点拆分成两个，这样与原来的效是相同的，但是更好分析并画出流程图，如

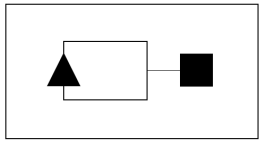


图16 同终点问题

可以变换为

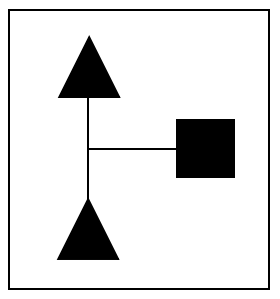


图17 例图（同终点问题）的第一次变换

然后拉直、伸缩，变换为

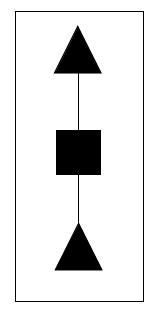


图18 例图（同终点问题）的第二次变换

可以看出，

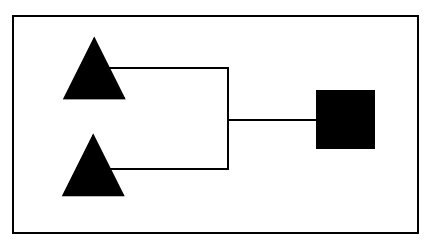
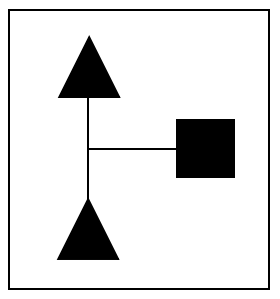
和

图19

都可以变换为

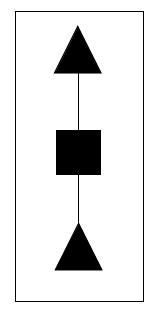


图20

故这两者的流程图是相同的，我们把变换后相同的两个地图成为**等价地图**。

### 2.5.4 同路问题

当两条路汇入同一条路时，我们不能将这条路拆分，因为车辆汇入同一条路时，这条道路的塞度会变大。

故我们**不能**将

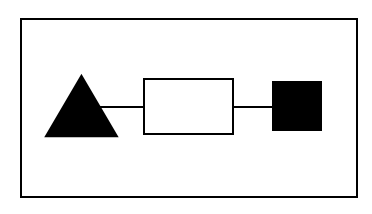


图21

变为

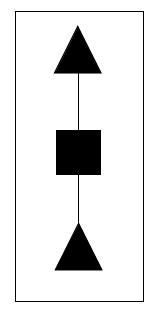


图22

可以看出，这两个地图不等价。

于是我们可以通过等价地图来方便我们制作流程图。

第三章 车辆路线预测和道路分析

3.1 路线预测

城市交通网络中，车辆的路线驾驶难以以数学模型进行有效建模。车辆行驶至岔路口时，驾驶员会依据自身行车经验、道路状况、天气因素等诸多原因而选择不同的路径。本节内容主要解决的问题便是如何准确高效地预测车辆行驶的途径。

本节选择蒙特卡洛树搜索的科研方法来模拟车辆道路选择，蒙特卡洛树搜索是通过汽车随机道路选择或简单策略的方法来模拟真实岔路选择，通过模拟结果更新蒙特卡洛决策树，最终使AI在面对真实选择时，可以根据决策树模拟成果做出最佳判断。

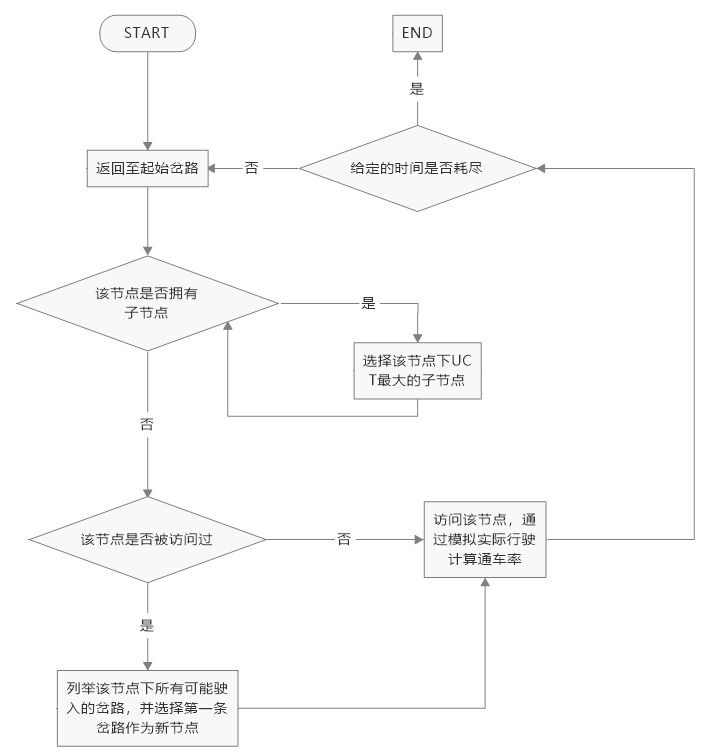


图23 蒙特卡洛树搜索

流程图演示的内容即为蒙特卡洛树搜索的运作机理。我们需要定义一个基础策略，选择当前岔路上最有可能被选择的子节点来模拟驾驶员面对不同道路时的选择，由此衍生出不同的路径，展开这个子节点，枚举出所有岔路驶入的可能性，基于这个子节点完成一次自博弈。我们对不同路径的通车率进行计算，记录自博弈结果并更新相关数据，反馈至蒙特卡洛信息树中，由此可以得到一条通车率最高的路径，获得最佳的选择。

对于车辆行驶途径的通车率，这里提出一个方法，我们不仅可以用车辆自身的历史堵车数据，也可以采用车辆途径道路的历史堵塞记录来丰富和完善UCT函数中通车率的计算，尽可能趋向现实生活，做出更合理的判断。

但是我们能紧紧依靠通车率来判断路径最佳或否吗？显然是不行的，例如某条道路在平时塞度很高，但某个大型活动，比如演唱会等大型活动将车辆吸引离开，导致这条道路偶然性地发生通车率较高的结果。同理，有些道路原本通车率很高，来往车辆较少，但是也可能因为交通事故等偶然原因而导致通车不畅，此时我们如果不修改通车率，蒙特卡洛模拟算法便会在这条发生偶然性错误的道路上投入大量搜索的力度，最终很大可能得到一个荒谬的结果。所以我们不能仅仅靠通车率来判断最优路径，而应该使用某个函数来修正它。这里我们使用UCT函数来确定蒙特卡洛模拟选择的对象。

以下为UCT函数：



其中表示驶入道路m的车辆数，表示通过道路m的车辆数（不发生堵车），表示通车率，表示总车辆数，表示修正函数（C为常数且C > 0），使AI选择时倾向于未被统计的路线，用来平衡通车率中蕴含的偶然性（值得一提的是，C的取值会影响AI对子节点的选择。C的取值越低，蒙特卡洛树搜索便越倾向于选择通车率更高的子节点，反之，C的取值越高，蒙特卡洛树搜索便越倾向于选择访问次数更少的子节点），这样每次AI选择时的通过选择UCT值最高的路径便是大多数情况中通车率最高的路径，随着模拟时间的增加，模拟次数的提高，蒙特卡洛树搜索的准确率会不断提高，算法的表现会更加优秀，其模拟出的结果也会更加趋于现实生活。

接下来，我们对蒙特卡洛树搜索流程图进行分析：

1. 首先我们需要判断当前节点是否拥有子节点（即拥有我们已经模拟出来的岔路，因为我们在之前访问该节点时，已经列举了所有的子节点，在未到达目的地之前，车辆不可能止步于此（忽略故障等偶然因素）所以我们应该尽可能地到达我们未访问的节点来分析该节点的通车率。
2. 若该节点拥有子节点，则选择该节点下UCT值最大的子节点（为什么选择UCT值最大的子节点，上文已经附有解释，这里不作赘述。）
3. 直到该节点下没有子节点，我们判断该节点是否被访问过，若没有则访问该节点，计算其通车率，更新蒙特卡洛信息树（这里是流程图最为核心的部分，我们在这里计算各个岔路的通车率，并反馈至上一级岔路以及整个蒙特卡洛信息树，用于最终的岔路决策。
4. 若该节点已经被访问过，我们应该列举该节点下所有可能驶入的岔路作为并选择第一个子节点作为新节点：我们在该节点已经计算通车率的前提下依然访问该节点，这证明该节点的理论价值相较于其他节点更高，我们便应该进一步访问该节点下的子节点，探索这条道路进一步通车的概率，以此来完善最终结果。
5. 判断时间是否耗尽。直到时间耗尽或者蒙特卡洛树搜索模拟出最终目的地之前，算法模拟将不断重复迭代。

3.2 蒙特卡洛算法优化环境相似性分析

传统蒙特卡洛算法中，模拟的过程仅会通过UCT函数的最大值来选择接下来的子节点，究其原因，蒙特卡洛树搜索仅仅依靠车辆本身的历史堵车经历、车辆途径道路的历史堵塞记录和地图的堵塞历史来模拟堵车的概率，而无法推算现实生活之中为什么会发生堵塞，怎么样才会发生堵塞。举个例子，每年暑假时，大量游客慕名涌入著名景点拍照打卡，造成交通的堵塞。而蒙特卡洛树搜索仅仅能够模拟出：经过特定的时间，在特定的地点，道路堵车率会有明显的提高，而不能推算出这是放暑假导致的结果。倘若我们人为的将暑假更换时间，蒙特卡洛模拟推算出的堵车率便自然不攻自破了。

为此，本节提出了蒙特卡洛算法的优化——结合环境的相似性进行分析，使蒙特卡洛树搜索推算出来的结果更加贴近于现实生活。

具体分析：从空间上推算，如果我们在现实生活中的某一条道路上遭遇长时间的堵车或连续性堵车，我们可以自然推算出：这附近可能有演唱会之类的大型活动导致范围性交通堵塞，从而提前选择另一条道路来避免这次堵塞。同理，在蒙特卡洛树搜索中，我们可以在分析访问节点时记录下节点的堵车次数和频率，后续模拟时，如果某一支路的堵车频率超过一定阈值时，我们便可以推算出该支路的大致情况：可能这一条支路上会出现大范围堵塞。这样我们便可以在之后的路线预测中，尽可能避免车辆从这条支路驶入（降低这些节点的修正函数的权重值，下文也是相同的做法），以此减少堵塞的概率。从时间上推算，如果我们在暑假计划出行，我们会尝试避免途经附近知名的景点，这是因为我们提前知道了车流的大量涌入大概率会导致景点附近发生范围性交通堵塞。同理，在蒙特卡洛树搜索中，我们可以预先设置好如寒暑假等可能造成车流量较大的时间节点。在实际应用时，我们可以提前判断使用时间与预先设计好的时间是否匹配。如果匹配，我们就应该设置蒙特卡洛模拟算法倾向于不要选择该时间下大概率堵车的支路。

## 3.3 塞度预测与道路设计建议

基于第一节的路线预测，我们得到了车辆行驶至目的地的最优路径，同理，我们稍微修改一下UCT函数，我们便可以得到有关于堵车率的计算和预测函数，即车辆行驶至目的地的最劣路径。随着在线推荐的普及，将有大量得到路线指引的车辆倾向于驶入通车率高的道路，这将会引起道路堵车率的提升，我们便可以对未来一段时间的大致塞度做粗略的预测。通过这份未来塞度预测的图纸，我们便能够帮助有关部门更加清晰地规划未来的道路设计和修建工程。

基于以上的分析，我们在这里提出几点建议：

1. 城市多条道路的交汇处通车率相较于其他岔路更低，我们认为可以在此对道路进行拓宽。
2. 在未来一段时间持续开展大型活动的场地附近，通车率也会相应降低，可以预先采用拓宽道路，加装红绿灯的方式来提高通车率，建立良好的交通秩序。

# 第四章 结论

本小组针对演唱会造成的交通问题，进行研究分析，得出以下结论：

1. 对道路的合理建设，可以有效缓解交通的堵塞问题，在此基础上推动城市经济文化的蓬勃发展，避免因流通问题影响城市的经济发展和高效率运行。
2. 在面对交通堵塞的问题时，运用蒙特卡洛的方法，可以进行对路线预测以及塞度预测，从而在道路建设上有更加明确的建设方向，避免因道路建设不当造成的交通堵塞问题。
3. 在面对现实中复杂的交通问题时，合理利用图像模拟，可以化繁为简，化抽象为具体，更加直观地面对问题。如：本研究中对道路进行变换，利用流程图来模拟理想状态下的道路，以便第三章中的深入研究。

4.数学方法的应用不止于理论区域，还可以在实际生活中有广泛用途，并且可以不断优化。如：蒙特卡洛在理论方面上，可以应用于求定积分，并且在天体物理学中，蒙特卡洛方法被广泛用于模拟星系演化和微波辐射穿过粗糙行星表面的情况。在实际应用方面上，蒙特卡罗方法被广泛应用于工程设计中的敏感性分析和定量概率分析等。

# 附：参考文献

1. 韩东轩.面向交通网络预测与评估的可视分析研究[D].西南科技大学,2021.DOI:10.27415/d.cnki.gxngc.2021.000571
2. 赵婷.城市公共交通多模式交通出行路径优化研究[D].重庆交通大学,2019.DOI:10.27671/d.cnki.gcjtc.2019.000163
3. 陈涛,陈森发.涨落后的城市道路交通拥挤蒙特卡洛预测[J].系统工程理论与实践,2004,(12):123-127.
4. 梯度世界.蒙特卡洛方法，用最粗暴的方式解最难的题[EB/OL].https://www.bilibili.com/video/BV1hV4y1Q7TR.2023-3-31
5. ClockSkew.蒙特卡洛方法概述[EB/OL].https://zhuanlan.zhihu.com/p/648985453.2023-08-09
6. 科普中国.蒙特卡洛法[EB/OL].https://baike.baidu.com/item/%E8%92%99%E7%89%B9%E5%8D%A1%E7%BD%97%E6%B3%95/1225057?fr=ge\_ala.2023-12-10