浙江大学

课 程 大 作 业 论 文

题目　动态交通环境下的辅助驾驶决策方法

所在小组　第五小组

指导老师　梁军、葛志强　教授

课程名称　运筹学与系统工程

所在学院　控制科学与工程学院

完成时间　**2019 年 06 月**

目 录

动态交通环境下的辅助驾驶决策方法

——以玉泉到紫金港的校车调度为例

第五小组

（浙江大学 控制科学与工程学院，浙江 杭州 310027）

摘 要：本文给出了一种面向动态交通环境的辅助驾驶决策方法，能够基于动态车流信息给出路线规划方案。为了实现这一点，本方案以各个路口为节点，将道路简化为离散化网格图；基于状态对环境进行了离散化建模，使计算机得以模拟动态交通环境；通过道路长宽，其他车辆数目的因素建模计算各节点之间的行驶代价，构建车辆运行规则。在上述建模基础上，方案使用 A*和强化学习两种决策算法分别实现了实时路线规划。

关键词：动态交通环境；辅助驾驶；A*；Q-Learning；决策方法；

Assistant Driving Decision-Making Method in Dynamic Traffic Environment——Take the School Bus Dispatching From Yuquan to Zijingang campus As An Example

The Fifth Group

(College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China)

Abstract: this paper presents an assistant driving decision-making method for dynamic traffic environment, which can provide route planning scheme based on dynamic traffic information. In order to achieve this, this scheme takes each intersection as a node and simplifies the road into a discrete grid diagram. The environment is discretized based on the state, so that the computer can simulate the dynamic traffic environment. By modeling the length and width of the road and the number of other vehicles, the driving cost between each node is calculated and the running rules of vehicles are constructed. On the basis of the above modeling, two decision algorithms, A* and reinforcement learning, are used to realize real-time route planning.

Key words: Dynamic Traffic Environment; Assisted Driving; A*; Q-Learning; Decision-making Method

0 引言

智能交通系统（Intelligent Transportation System，ITS）旨在构建更加安全，舒适，稳定的交通环境；车辆高级驾驶辅助系统（Advanced Driver Assistance System，ADAS）近年来也有了较快的发展[1]。其中，行车环境信息采集、行车环境表征建模及驾驶行辅助决策是驾驶辅助决策的重要研究方向[2-4]。本文主要关注辅助驾驶决策中基于状态信息转换的驾驶路径生成研究，旨在探讨动态交通环境下的车辆调度决策方法。

目前关于出行的导航规划问题，大多是基于静态道路信息进行驾驶员的辅助驾驶路线决策，主要侧重于图论中的路径规划。涉及车辆出行规划的研究大致可分为两类[5]：第一类主要为车辆集群推荐出行规划方案[6-8]，针对车队整体运营目标进行优化，不反映驾驶员的驾驶需求。第二类问题针对单辆汽车，考虑最短距离[9]、最短时间[9,10]或最短能量消耗[11]等目标，为驾驶员推荐最优的驾驶路线，但仅考虑了确定性环境。

现有的地图导航软件，如百度地图、高德地图等，其动态信息大多是基于历史数据库的迭代更新，利用数据驱动的方法，针对交通流这一批次性的间歇过程进行道路交通的建模，并没有真正考虑到动态交通环境所带来的道路车辆转移规则，难以为驾驶员提供综合覆盖区域动态信息等因素的辅助驾驶决策结果。

根据文献[12]的总结，无人驾驶车辆行为决策方法可以根据驾驶策略搜索和存储方式的不同，分为四类：状态机模型、决策/行为树模型、基于知识的推理决策模型和基于效用/价值的决策模型。本文采用了状态机模型的方法，即在司机驾驶的过程中，根据当前实时的交通环境状态进行辅助决策，通过构建连通图来描述不同的驾驶途经点以及状态之间的转移关系，从而根据驾驶状态的迁移地生成驾驶策略。此外，本文重点关注初始道路交通全局信息已知情况下的动态交通流建模和车辆路径辅助决策方法，将采用基于状态更新的离散模型进行建模，并利用 A*和 Q-Learning 两种决策算法进行求解，给出动态交通环境下的辅助决策结果，为玉泉到紫金港的校车进行优化调度。

1 问题描述

1.1 问题目标重述

本文关注在拥有区域交通流的全局信息条件下，如何考虑动态交通流的变化情况，对玉泉开往到紫金港的校车司机提供一种辅助性的驾驶路线决策方法。

首先，限定动态交通区域为玉泉到紫金港可能经过部分区域，对道路进行一定的简化，考虑车流量大的主干道，删减部分小路，以降低模型的复杂度且不失一般性。同时，限定该区域和外界没有车辆交换，仅考虑区域内的车辆流动和转移。

其次，对车流分布以及道路交通规则进行数学建模，考虑道路车辆密度、车道宽度、道路长度、目的地导向信息等因素，构建车辆状态转移模型和区域交通流动模型。

最后，基于建立的交通规则模型，对从玉泉开往紫金港方向的校车，以朝向目标点、避免拥堵等为综合目标，以不同车流分布（动态变化）和道路交通规则为约束，寻求一种代价最小的行车方案。

1.2 形式化定义

辅助驾驶决策问题可以结合地图构建、路口标号，将其转化为数学问题，而优化目标为寻求一条最优的路径。其中，最优的定义是使得总代价最小，而代价与多种因素相关。

为了简便起见，需要将道路进行简化，选取其中靠近主干道的部分路口，同时假设路口见道路呈直线（事实上也基本是直线），最终将道路简化成如图 1 所示的网格化地图。

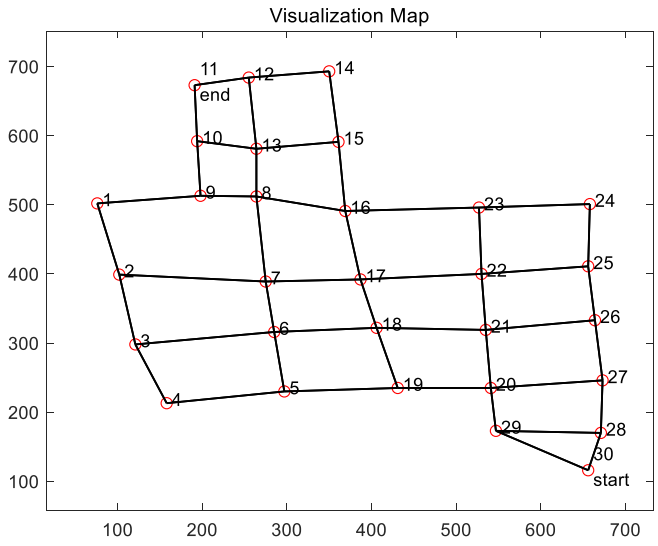

图 1 玉泉-紫金港交通范围内的简化网格道路

基本假设：

1. 区域内一共有 30 个路口，依次编号为 $p_1 \sim p_{30}$，即 $P=\{p_1, p_2, \cdots, p_{30}\}$；
2. 校车运行的起点为玉泉校区 p_{30}，终点为紫金港校区为 p_{11}；
3. 不考虑区域边界的车辆流动，即假设该区域是封闭的，车辆只在图中道路上行驶。

优化目标：本文辅助驾驶决策方法的形式化优化目标为 $\min Cost=\sum_{i=1}^{30}\sum_{j=1}^{30}C_{ij}$，其中 C_{ij} 为路口代价函数，即使得从起始点到目的地车辆运行的总代价最小。

1.3 论文思路说明

为了更好地阐述动态交通环境下的辅助驾驶决策方法的总体思路，需要对论文的思路进行分析和说明，论文总体框架和思路说明如图 2 所示。

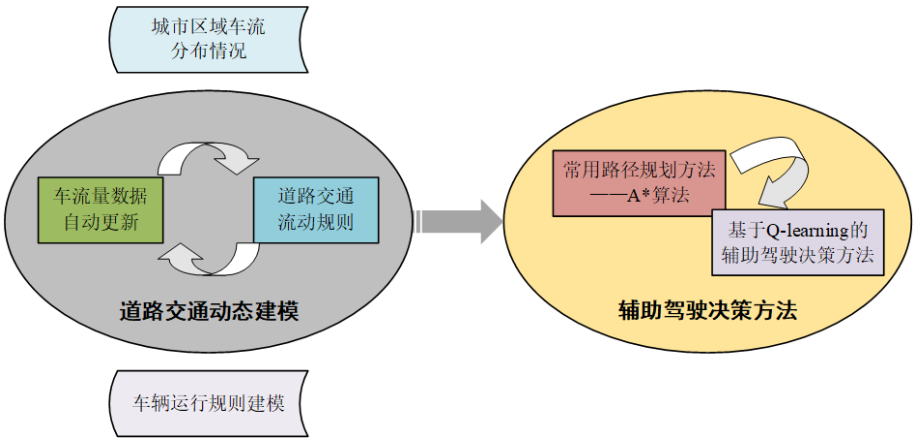

图 2　论文总体框架和思路说明

图？描述的论文架构思路，总体而言，可以分为道路交通动态建模和辅助驾驶决策方法这两个部分。

在道路交通动态建模中，首先根据车流热力图获取城市区域车流分布情况的初始值，然后构建车辆运行规则的模型，通过道路动态变化情况来驱动该模型。我们借鉴了元胞自动机的思想，对交通流进行动态模拟，其中，车流量数据的更新和道路交通流动规则之间会随着程序运行而相互迭代更新。

在辅助驾驶决策方法部分，分别考虑全局信息视角和局部信息视角下的辅助驾驶路线决策，采用了 A*和 Q-Learning 两种不同的算法。

2　车辆运行规则与道路交通建模

2.1　城市区域车流分布情况

2.1.1　道路信息提取与简化处理

为方便后续进行道路交通建模与决策，首先通过百度地图获取从玉泉校区到紫金港校区的地图全貌，如图 3 所示。

图 3　玉泉到紫金港范围内的地图全貌

注意到，在图 3 中，除主干道以外，还有很多道路小分枝可以选择。在此，为了将侧重点放在建模与决策方案上，忽略掉道路小分枝，并以主干道为重点考虑的交通道路，从而完成对地图的简化。

在原地图的基础上，将非道路部分涂色为黑色，可以得到下图所示的地图图片，作为道路信息提取的输入，如图 4 所示。

图 4　初步预处理后的地图图片

由于图 4 将大部分非道路区域都进行了黑色染色处理，从而可以突出道路部分。因此，将上图转化为灰度图片，并根据每个像素的灰度值作滤波处理，得到灰度化后的地图图像，如图 5 所示。

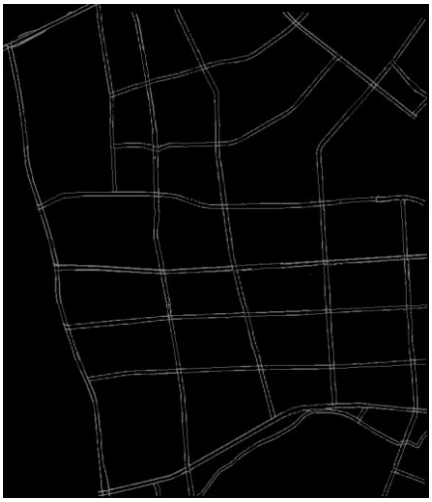

图 5　灰度化后的地图图片

将上述灰度图进行拓展并卷积，使得在卷积核范围内出现灰度值>0 的像素时令该卷积核范围的像素为白色（即可通道路），进行滤波处理后得到二值栅格化地图，如图 6 所示。

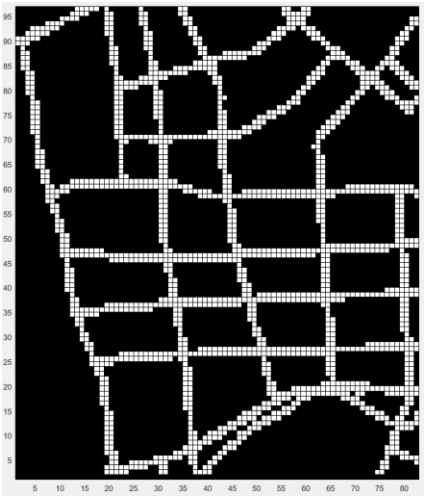

图 6　二值栅格化地图

由图 6 可以看出，该二值栅格化地图很好地体现了原始地图中的道路信息。为便于进行离散道路建模，将上述二值栅格化地图抽象为一个无向图。其中，每一个路口抽象为一个节点，路口之间若有道路连通，则将道路抽象为边，得到抽象的无向图如图 7 所示。

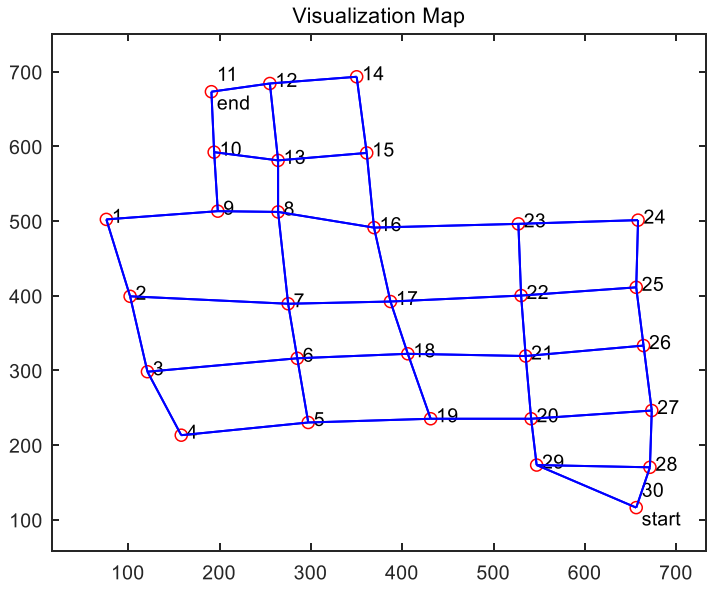

图 7　简化提取后的道路区域无向图

其中，每个路口的横纵坐标取决于其在图片中的像素位置。由于所有路口统一由同一张图片标定位置，其尺度是一样的，因而该无向图能够很好地表征原来的道路信息图。至此，对道路信息的提取与简化预处理完成。

2.1.2　基于交通热力图的车流分布

查阅资料知，百度地图[1]提供了关于实时路况查询的 API 接口，其中，矩形区域实时路况查询的 API 示例如表 1、表 2 所示。

[1] 百度地图 API http://lbsyun.baidu.com/index.php?title=webapi/traffic

表 1 百度地图 API 请求参数（节选）

API 参数名称	参数含义	参数类型	备注
bounds	矩形区域，左下角和右上角的经纬度坐标点	string	坐标点顺序为"左下;右上"，坐标对间使用;号分隔，格式为：纬度,经度;纬度,经度。对角线距离不超过 2 公里。如：39.912078,116.464303;39.918276,116.475442
road_grade	道路等级	int	用户可进行道路等级筛选，支持选择多个道路等级。道路等级之间使用英文“,”分隔。 默认值：road_grade=0 道路等级对应表如下： 0：全部驾车道路 1：高速路 2：环路及快速路 3：主干路 4：次干路 5：支干路

表 2 百度地图 API 返回参数（节选）

API 参数名称	参数含义	参数类型	备注
status	路段拥堵评价	int	支持以下值： 0：畅通 1：良好 2：缓行 3：拥堵 4：严重拥堵
road_name	道路名称	string	如：“信息路”、“北五环”

由上述 API 接口可以根据经纬度得到各道路的路段拥堵评价，该评价分为 5 大类，对应关系为：0-畅通，1-良好，2-缓行，3-拥堵，4-严重拥堵。借用如上 API 接口，提出一种获取地图车流密度分布的想法：

① 根据从玉泉到紫金港范围内各道路所处的经纬度，将道路按一定经纬度步长切分为道路对应的经纬度点，并通过 API 返回该点的拥堵评价

② 利用整个玉泉到紫金港范围内各点的拥堵评价，做出道路拥堵评价分层热力图

③ 根据各道路在热力图中的拥堵情况，利用以下平均公式计算出该道路的总体车流密度指标：

$$\text{Traffic Density} = \frac{\sum_{i=1}^{n} c_i l_i}{\sum l_i}$$

其中，c_i为该道路中第i个经纬度点对应的道路拥堵评价值，l_i为该道路的经纬度步长，$\sum l_i$为该道路的总长度。

按照上述想法，得到地图道路拥堵评价分层热力图，如图 8 所示。

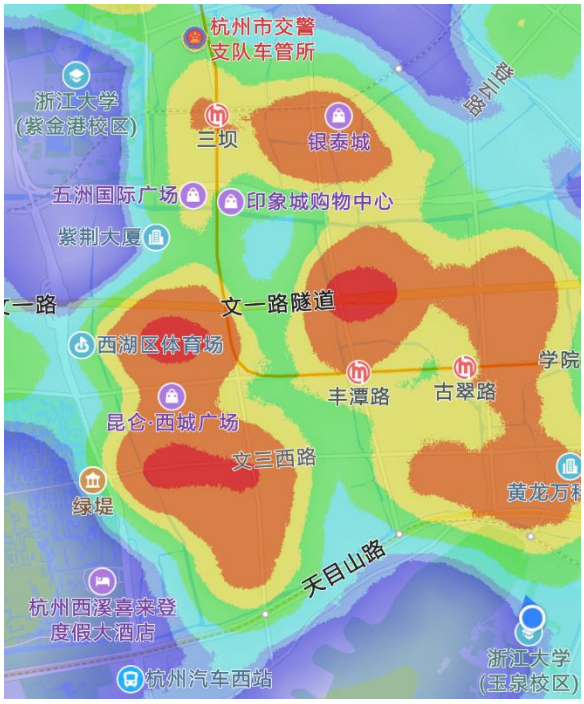

图 8　地图道路拥堵评价分层热力图

如图 8 所示，青色代表 0，绿色代表 1，黄色代表 2，橙色代表 3，红色代表 4；紫色代表未知。根据计算，得到地图道路车流密度上三角矩阵如图 9 所示。

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1		2.72							2.73																					
2			2.84				2.23																							
3				2.02		2.19																								
4					2.27																									
5						2.49													2.74											
6							3.19											2.41												
7								2.12									2.07													
8									2.75				2.68			2.97														
9										2.84																				
10											2.87		2.27																	
11												2.41																		
12													2.91	2.43																
13															2.71															
14															2.69															
15																2.96														
16																	2.55						2.12							
17																		2.65				2.35								
18																			2.62		2.30									
19																				2.13										
20																					2.32					2.13		2.22		
21																						2.62				3.00				
22																							2.43		2.54					
23																								2.24						
24																									2.19					
25																										2.39				
26																											2.36			
27																												2.82		
28																													2.92	3.23
29																														2.58
30																														

图 9　地图道路车流密度上三角矩阵

至此，完成了对初始交通状态信息的读取与整理，并得到了地图道路车流密度的上三角矩阵。

2.2　道路交通动态建模

2.2.1　动态交通环境的构建思路

为了验证决策方法在动态交通下的运行效果，我们需要对动态交通进行基于状态的离散建模。已知紫金港-玉泉区域共有 n 个路口，根据实际交通网络状况获得模型基本变量信息

如下：

路口车辆数矩阵：$X_{n\times1}$

道路车流密度矩阵：$\rho_{n\times n}$

道路长度矩阵：$Y_{n\times n}$

道路宽度矩阵：$\alpha_{n\times n}$

其中，每条道路由其两端路口标号 i, j 表示，矩阵 $Y_{n\times n}, \alpha_{n\times n}$ 为固定值，由前期对实际道路抽象建模获取，$X_{n\times1}, \rho_{n\times n}$ 为实时更新变量，通过交通流动态模型得到每次迭代的路口间车辆转移信息，进而更新为各路口、道路车辆转移后的状态信息，具体过程如图 10 所示。

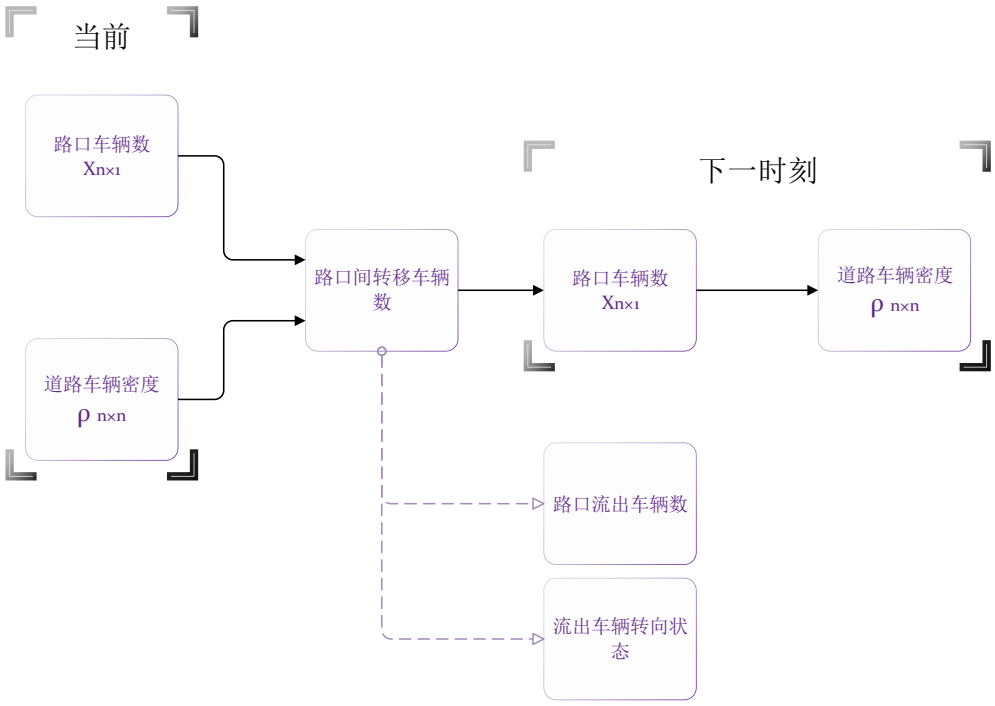

图 10　动态车流更新过程

每次交通流更新时，将当前路口、道路车辆信息投入动态交通网络模型，得到本次路口间转移车辆信息，包括路口流出车辆数矩阵 $L_{n\times1}$，流出车辆转向状态矩阵 $P_{n\times n}$。基于马尔可夫过程，其后继状态 S' 完全由当前状态 S 决定，车流量后继更新状态：

$$X_{i(t+1)} = X_{i(t)} - L_{i(t)} + \sum_{j=1}^{n} L_{j(t)} * P_{j,i(t)}$$

$$\rho_{i,j(t+1)} = 0.5 * X_{i,t+1} + 0.5 * X_{j,t+1}$$

2.2.2 动态交通环境的建模过程

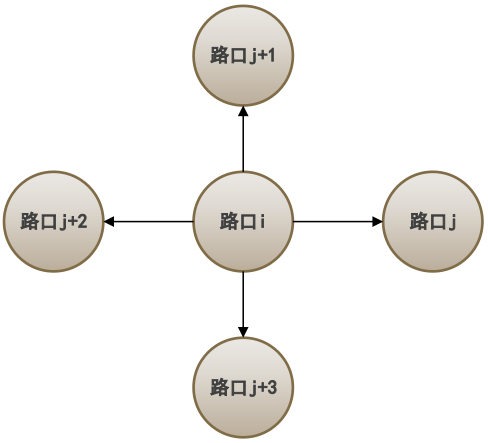

图 11 路口建模示意图

为模拟路口间车辆流动情况，将单一路口流出车辆去向转换为概率问题。通过构造一定规则，确定路口转向的概率模型，并对所有从该路口转出概率值做归一化处理，以满足所有路口转出概率和为 1。

$$\sum_{j=1}^{n} P_{i,j} = \sum P_{i,j} + P_{i,j+1} + P_{i,j+2} + P_{i,j+3} = 1$$

（1）路口转向概率模型

$$\text{转向规律}\begin{cases}\text{一般性}\begin{cases}\text{车辆密度}=\rho_{i,j}\uparrow,\ P\uparrow\\ \text{主干程度}\alpha_{i,j}\uparrow, P\uparrow\end{cases}\\ \text{特殊性}\begin{cases}\text{避障性，}\ \text{行驶速度}\left(V_{max}-\dfrac{V_{max}}{\rho_{max}}\rho_{i,j}\right)\uparrow,\ P\uparrow\\ \text{目的地随机性}\end{cases}\end{cases}$$

由于该动态交通网络模拟一般车流运行状态，其目的地未知且带有随机性，在构建路口转向概率模型时，综合一般性思维与特殊情况考虑，对路口车辆转向选择做出合理推测。基于一般性思维考虑，对一目的地未知的车辆，其驶入主干道、车流量大的道路概率明显高于其他道路，即 $\rho_{i,j}\uparrow$、$\alpha_{i,j}\uparrow$，则$P\uparrow$,。

基于特殊情况考虑，由于车辆目的地未知；若对该车，在此路口转向方式不唯一时，可能基于避障思想，选择车流量较小的道路。即$\left(V_{max}-\dfrac{V_{max}}{\rho_{max}}\rho_{i,j}\right)\uparrow$、$\rho_{i,j}\downarrow$，则$P\uparrow$。反之，若对该车，在此路口转向方式唯一且确定时，转向选择仅有目的地为唯一导向，模型中由随机数模拟产生。

综合上述因素考量，对路口转向概率模型：

$$P_{i,j}=\alpha_{i,j}(\beta_1*\rho_{i,j}+\beta_2(V_{max}-\frac{V_{max}}{\rho_{max}}\rho_{i,j})+\beta_3 e^{\Delta x_{i,j}}+\beta_4 rand)$$

（2）路口流出车辆模型

$$\text{流出车辆}\begin{cases}\text{车辆密度}\\ \text{车辆行驶速度}\begin{cases}\text{基本行驶速度}\\ \text{受拥塞影响速度}\end{cases}\end{cases}$$

基于理论建模：路口流出车辆数=道路宽度×车辆密度×车辆速度×时间，本模型为基于状态的离散模型，每次迭代更新周期时间间隔为 1，其余变量均可由模型基本参数获取，路口流出车辆数：

$$L_j=\alpha_{i,j}\rho_{i,j}(V_{max}-\frac{V_{max}}{\rho_{max}}\rho_{i,j})$$

2.3 车辆运行规则建模

为了进行路径规划，需要根据实际情况，对车辆运行代价进行建模，根据已知的道路与路口信息建立起车辆从一个路口到另一个路口的转移代价函数。

2.3.1 变量定义

ρ：n×n，车流密度矩阵，ρ_{ij} 表示第 i 个路口和第 j 个路口之间的连通道路的车流密度；

Y：n×n，道路长度矩阵，y_{ij} 表示路口之间的路的长度；

X：n×1，路口车辆数矩阵，x_i 表示每个路口的车辆数；

A：n×n，可达矩阵，a_{ij} 表示路口 i 与路口 j 是否可达，可达为 1，不可达为 0。

2.3.2 建模思路

我们将运行代价等同于花费的时间，而路口转移的花费时间不仅需要考虑转移过程中的时间（时间越长代价越大），还需要考虑转移之后去往目的地的时间花费（距离越远代价越大）。因此考虑路口转移代价由以下三部分组成。

$$\text{路口转移代价}\begin{cases}\text{在两路口间相连道路的行驶时间}\\ \text{路口等待时间}\\ \text{目标路口到终点的行驶时间}\end{cases}$$

2.3.3 建模过程

根据上述三部分组成分别建立模型。

（1）路口间道路行驶时间建模：两路口间相连道路行驶时间=$\frac{\text{道路长度}}{\text{行驶速度}}$

已知道路密度矩阵 ρ 和道路长度矩阵 Y，假设道路密度与行驶速度呈线性关系，可以建立速率与密度的函数关系式：$V_{\max}-\frac{V_{\max}}{\rho_{\max}}\rho_{ij}$ ，因此路口间道路行驶时间 t_1 的表达式为：

$$t_1=\frac{y_{ij}}{V_{\max}-\frac{V_{\max}}{\rho_{\max}}\rho_{ij}}$$

其中 $V_{\max}$ 表示道路速度上限，$\rho_{\max}$ 表示密度上限，对应速度为 0。

（2）路口等待时间建模：假设路口等待时间与路口车辆数呈线性关系，则两者的关系式为路口等待时间= $k\cdot$路口车辆数，已知路口车辆数矩阵 X，因此道路等待时间 t_2 的表达式为：　　$t_2=k\bullet x_i$

（3）目标路口到终点的行驶时间：目标路口到终点的行驶时间=$\frac{\text{距离}}{\text{行驶速度}}$

其中两点距离可以根据目标路口的坐标和终点的坐标计算得出，假设行驶速度即为当前道路行驶速度，因此可得目标路口到终点的行驶时间 t_3 的表达式为：

$$t_3=\frac{d}{V_{\max}-\frac{V_{\max}}{\rho_{\max}}\rho_{ij}}$$

由于以上三部分函数的量纲相同，因此总代价函数 C_{ij} 为上述三部分的线性函数，可得从第 i 个路口到第 j 个路口的转移代价表达式如下：

$$C_{ij}=k_1\bullet\frac{y_{ij}}{V_{\max}-\frac{V_{\max}}{\rho_{\max}}\rho_{ij}}+k_2\bullet x_i+k_3\bullet\frac{d}{V_{\max}-\frac{V_{\max}}{\rho_{\max}}\rho_{ij}}$$

以上表达式仅考虑了两个相邻路口的转移代价，而对于两个不相邻的路口，希望其转移代价为 ∞，因此在以上整个表达式外乘上可达矩阵元素的倒数，即可表示任意两个路口间的转移代价，如下：

$$C_{ij}=(k_1\bullet\frac{y_{ij}}{V_{\max}-\frac{V_{\max}}{\rho_{\max}}\rho_{ij}}+k_2\bullet x_i+k_3\bullet\frac{d}{V_{\max}-\frac{V_{\max}}{\rho_{\max}}\rho_{ij}})\cdot\frac{1}{a_{ij}}$$

3 辅助驾驶决策方法

3.1 动态交通环节下的决策算法选取

基于对上述动态交通流模型的分析，问题已转化为在无向连通图模型下，求解起点至终点的最小代价路径，并要求合理使用全局或局部信息。因此，选用 A*算法和强化学习算法，分别使用全局信息和局部信息进行路径规划。

3.2 常用路径规划方法——A*

3.2.1 算法原理概述

A*算法[13]（又称 A Star 算法）是一种静态路网中求解最短路十分有效的直接搜索方法。其核心思想可由如下公式表征：

$$f(n)=g(n)+h(n)$$

n：地图中某一节点；

$f(n)$：从初始点到目标点，并经过节点 n 的代价估计函数；

$g(n)$：从初始节点到 n 节点的实际代价，即起始节点到当前节点的实际代价；

$h(n)$：从 n 到目标节点最佳路径的估计代价。即当前节点到目标节点的估计代价。

从上式中可见，A*是一种典型的启发式寻路算法，其效果依赖于启发函数 $h(n)$ 的选择，即是否能恰当地选择对当前节点到目标节点的代价估计算法，决定了 A*算法的执行效果。$h(n)$ 的常用选择有当前节点到目标节点的欧式距离等。

3.2.2 算法流程说明

A*算法的算法流程如下：

a. 把起始节点添加到开启列表。

b. 主要循环体：

a) 寻找开启列表中 f 值最低的节点，选定他它为当前节点。

b) 把它加入到关闭列表。

c) 满足以下任一条件则终止循环，否则进入步骤 d)

* 终点被添加进了关闭列表，这时候路径被找到。

* 开启列表已经空了。这时候，路径不存在。

d) 对与当前节点的所有相邻节点

* 如果它不可通过或者已经在关闭列表中，略过它。

* 如果它不在开启列表中，把它添加进去。把当前节点作为这一节点的父节点。记录这一节点的 h , g 和 f 值。

* 如果它已经在开启列表中，用 g 值为参考检查是否有经过当前节点到此节点

的更小代价路径。如果是，就把这一节点的父节点改成当前节点，并且重新计算这一节点的 g 和 f 值。

c. 从目标节点开始，沿着每一节点的父节点移动直到回到起始节点，此即为算法寻到的路径。

3.2.3 算法适用范围

为了将 A*算法应用于本文讨论的问题，我们使用了以下改进方法。

（1）使用每一时刻的全局信息进行实时更新

A*算法要求地图为静态路网，但是本文对交通环境的建模为动态车流模型，故使用 A*算法在模型下求解最优路径时，每次使用区域内的全局信息规划出一条全局路径，沿此路径行进一步后，由于交通环境数据已发生演变，再次使用新的全局信息规划路径，重复这个过程，完成了 A*在动态环境下的寻路。

这种方法之所以可行，得益于模型所取的区域不太大，因此每一次全局路径规划的时间复杂度并不高，故得以实时使用 A*算法更新路径。

（2）从终点做反向传播构成启发函数 $h(n)$

同样得益于地图复杂度不高，我们采用自终点开始向全局进行反向传播的方法，得到了全局任意一点到终点的最短路径，以此为 A*算法中的启发函数 $h(n)$ 。

基于以上两点改进，我们实现了使用 A*算法的辅助交通决策，其效果将在下文中给出。

3.3 基于 Q-learning 的辅助驾驶决策方法

3.3.1 考虑局部信息的决策方法

如前文所述，在全局信息可观的情况下，A*算法是一种十分优秀的路径规划算法。但大多数实际情况下，全局信息对于 agent 而言是不可观的，这就导致 A*等一系列启发式搜索算法难以得到应用。如何解决全局信息不可观情况下的路径搜索策略，是一个亟待考虑的问题。

在本次问题求解中，我们主要采用了基于强化学习的路径规划算法。在问题求解过程中，假若 agent 仅能获取局部信息，如图 12 所示，当 agent 处于路口 $Cross\,1$ 时，其可观的信息仅包括相邻路口 $Cross\,i$ 的车辆信息及相连道路 $Road(1,i)$ 的车流密度信息，强化学习方法的优势便能够得到很好地体现。

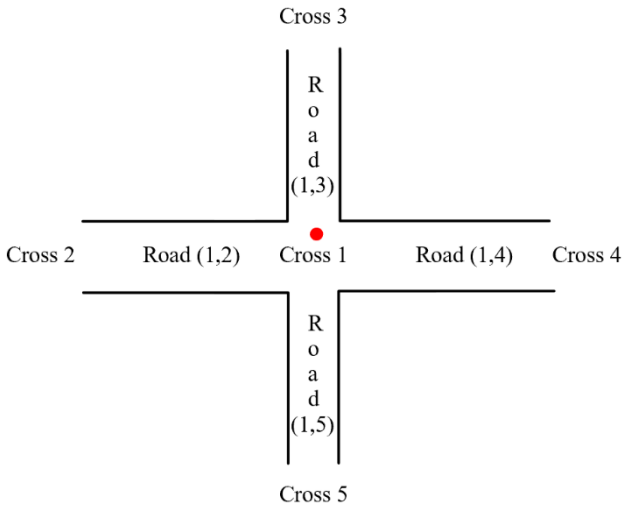

图 12　局部信息可观示意图

3.3.2　符号定义

状态 s：agent 用于决策下一时刻所需采取的策略所使用的信息。约定 t 时刻的状态记为 s_t，整个问题的动作集为 $S=\{s_1,s_2,...,s_m\}$。

动作 a：agent 在不同状态下所采取的策略，通过采取不同的动作，agent 能够在不同的状态之间变换。约定 t 时刻的动作记为 a_t，整个问题的动作集为 $A=\{a_1,a_2,...,a_n\}$。

观测 o：agent 在不同时刻、不同状态下观测到的环境信息。约定 t 时刻的观测记为 o_t。

策略 π：agent 在状态 s_i 下采取动作 a_j 的概率，即 $\pi(a_j \mid s_i)=P(A_t=a_j \mid S_t=s_i)$。对于确定性过程，对于状态 s_i 应有唯一的一个最优策略，即：

$$\pi(a_j \mid s_i)=\begin{cases}1 & j=j^* \\ 0 & else\end{cases}$$

状态转移矩阵 P：状态转移矩阵的每一行对应一个状态，每一列对应另一个状态，元素 p_{ij} 代表由状态 s_i 到达状态 s_j 对应的概率 $P(s_j \mid s_i)$。显然，状态转移矩阵有性质 $\sum_{j=1}^{n} p_{ij}=1$

$$P=\begin{pmatrix} p_{11} & \cdots & p_{1n} \\ \vdots & \ddots & \vdots \\ p_{m1} & \cdots & p_{mm} \end{pmatrix} \quad m:number\ of\ states$$

代价值矩阵 R：代价值矩阵的每一行对应一个状态，每一列对应一个动作，元素 r_{ij} 代表在状态 s_i 下采取动作 a_j 对应的代价值。

$$R=\begin{pmatrix} r_{11} & \cdots & r_{1n} \\ \vdots & \ddots & \vdots \\ r_{m1} & \cdots & r_{mn} \end{pmatrix} \quad m:number\ of\ states; n:number\ of\ actions$$

折扣因子 γ：折扣因子 γ 表示表示未来奖赏与当下奖赏之间的相对重要程度。

回报 G_t：$G_t=R_{t+1}+\gamma R_{t+2}+\gamma^2 R_{t+2}+...=\sum_{k=0}^{\infty}\gamma^k R_{t+k+1}$，表示对未来奖赏值的加权和。

动作值矩阵 Q：代价值矩阵的每一行对应一个状态，每一列对应一个动作，其中元素 $q_{ij}=q(s_i,a_j)=E_\pi[R_{t+1}+\gamma q(s_{t+1},a_{t+1})\,|\,s_t=s_i,a_t=a_j]$ 代表在状态 s_i 下采取动作 a_j 对应的动作值期望。需要注意的是，动作值矩阵与代价值矩阵是两个不同的矩阵，准确地说，动作值矩阵是通过代价值矩阵进行迭代学习，最终收敛得到的矩阵，而最终的策略制定也是基于动作值矩阵进行的。

$$Q=\begin{pmatrix} q_{11} & \cdots & q_{1n} \\ \vdots & \ddots & \vdots \\ q_{m1} & \cdots & q_{mn} \end{pmatrix} \quad m:number\ of\ states;\ n:number\ of\ actions$$

3.3.3 强化学习算法介绍

（1）强化学习介绍

强化学习是本质上是一种 agent 与环境交互，并逐步学习不同状态下的最优策略的过程，其整体架构如图 13 所示。

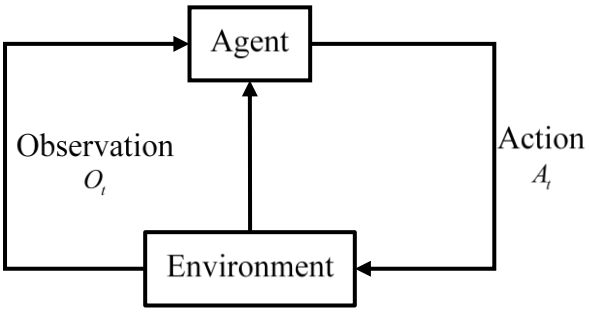

图 13 强化学习框架示意图

简单地说来，强化学习是一种无监督的学习，环境并不会直观地告诉 agent 所采取策略的优劣对错，而仅仅是通过一个奖赏值来帮助 agent 进行评估；同时，在 agent 与环境的交互过程中，其采取的策略也会影响环境。通过在逐步的迭代学习过程中，agent 执行更优的策略能够得到更大的奖赏值(或更小的代价值)，从而能够通过奖赏值逐步引导 agent 选择更优的策略。其次，由于 agent 需要在与环境的多次交互中“试错”性地进行学习，故而强化学习过程的反馈并非是瞬时的，而往往是具有时延的。

在学习过程中，agent 需要逐步建立起环境模型以估计环境的变化。一个模型需要包括转移概率 $P_{ss'}^a=P(S_{t+1}=s'\,|\,S_t=s,A_t=a)$ 与奖赏值估计 $R_s^a=E(R_{t+1}\,|\,S_t=s,A_t=a)$ 两个部分；其中，$P_{ss'}^a$ 表示在 $S_t=s,A_t=a$ 下转移到 $S_{t+1}=s'$ 的概率值，R_s^a 表示在 $S_t=s,A_t=a$ 下对于下一时刻奖赏值的估计。$P_{ss'}^a$ 通常可以通过蒙特卡洛随机模拟获得，而 R_s^a 通常通过对马尔科夫链的多次迭代收敛得到。

（2）马尔科夫决策过程(MDP)

强化学习的求解大多是基于 MDP 来实现的，为了方便后续更好地叙述算法原理，这里对 MDP 做简要的叙述。MDP 可以采用一个元组 $\langle S,A,P,R,\gamma\rangle$ 来进行表示，S 为有限状态

集合， A 为有限动作集合， P 为状态转移矩阵， R 为奖赏值矩阵， γ 为折扣因子。

MDP 的求解时基于 Bellman Expectation Equation 实现的，该方程为 MDP 的动作值矩阵迭代收敛提供了数学化的表示。对于每一个状态与动作构成的二元组 $\langle s,a\rangle$ ，可以通过下式来对动作值进行迭代更新：

$$q_\pi(s,a)=R_s^a+\gamma\sum_{s'\in S}P_{ss'}^a\sum_{a'\in A}\pi(a'\mid s')q_\pi(s',a')$$

随着迭代过程的进行，动作值矩阵 Q 会逐步收敛，而此时我们可以采用贪心的策略选择策略，也即选择状态 s 下 $q(s,a)$ 最大的动作 a 作为策略，即 $a=\arg\max_{a\in A}q(s,a)$ 。

考虑到 MDP 中 agent 决策与环境变化的强耦合性，MDP 中一个十分重要的问题是 agent 在决策过程中“开发”(exploration)与“利用”(exploitation)之间的权衡。Exploration 的意义在于 agent 能够去探索未曾涉足的环境，以获得更为全面的信息，但同时由于未知环境的信息不可观，exploration 可能导致 agent 到达不利的环境中。另一方面，exploitation 的意义在于 agent 能够利用之前已经学习到的知识进行决策，以获得当下最优的结果，但同时由于对整体环境信息的缺乏，agent 容易陷入到局部最优解中。

（2）Q-Learning 原理概述

考虑基于局部信息的路径规划问题，可以将其抽象为一个包含 30 个状态与 30 个动作的马尔科夫决策过程，本问题中我们采用 Q-Learning 的方法来求解此马尔科夫决策过程。

Q-Learning 是一种基于动作值矩阵来进行决策的强化学习方法，同样也是一种基于模型的强化学习方法。首先，我们需要基于蒙特卡洛随机模拟获取整个环境的转移概率 $P_{ss'}^a=P(S_{t+1}=s'\mid S_t=s,A_t=a)$ ，其基本流程如下：

Function Model Estimation

Initialize parameters $m(y,\langle s,a\rangle)=0$ $\#m(y,\langle s,a\rangle)=0$ *is used for counting the time which state y is been invited through state-action pair* $\langle s,a\rangle\in\langle S,A\rangle$

For each state-action pair $\langle s,a\rangle\in\langle S,A\rangle$：

For i=1,2,…,n **do**

For a state-action pair $\langle s,a\rangle$, simulate the following state $y\sim P(\bullet\mid\langle s,a\rangle)$ based on corresponding state-transition probability $P(\bullet\mid\langle s,a\rangle)$.

$m(y,\langle s,a\rangle)=m(y,\langle s,a\rangle)+1$

End for

$\forall y\in S,\hat{P}(y\mid\langle s,a\rangle)=\dfrac{m(y,\langle s,a\rangle)}{n}$ *# Estimate model through Monte-Carlo Simulation*

End for

Return $\hat{P}$

在获取到了环境模型之后，可以即可采用 Q-Learning 算法来进行策略选择。在强化学

习中，又有 agent 只能获取到一系列的动作值，所以其在采取动作时不仅要考虑到执行当下动作的奖赏，还需要考虑到执行这一动作后到达下一状态爱可能带来的未来的奖赏。这一点，用数学语言可以表示为：

$$q_\pi(s,a) = E_\pi[R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a]$$

所以，可以看到，Q-Learning 迭代需要涉及到的量包括：

- 某一特定状态 $S_t = s$ 下执行动作 $A_t = a$ 对应的奖赏值 R_{t+1}
- 某一特定状态 $S_t = s$ 下执行动作 $A_t = a$ 到达的下一状态 S_{t+1}
- 在状态 S_{t+1} 下依据当下策略 π (exploitation)/随机选择动作(exploration)执行动作 A_{t+1} 对应的 Q 值 $q_\pi(S_{t+1}, A_{t+1})$

首先，对于奖赏值 R_{t+1}，在本次问题求解中，我们采用每个路口走每条道路的代价值作为奖赏值。对于 agent 所处路口 $Cross\,i$，其仅知道其周围相连道路 $Road(i,j)\ j \in accessible\ to\ cross\ i$ 的代价值。

其次，关于 agent 在特定状态 $S_t = s$ 下执行动作 $A_t = a$ 到达的下一状态 S_{t+1}，这一点及需要采用之前得到的估计模型来进行求解。估计模型能够给出在状态 $S_t = s$ 下执行动作 $A_t = a$ 到达不同 $S_{t+1} = s'$ 的概率 $\hat{P}(s' \mid \langle s, a \rangle)$，从而可以在获得 agent 的下一状态。

再则，关于 agent 在状态 S_{t+1} 下依据当下策略 π (exploitation)/随机选择动作(exploration)执行动作 A_{t+1} 对应的 Q 值 $q_\pi(S_{t+1}, A_{t+1})$。由于 Q-Learning 是一个迭代的过程，在对 $q_\pi(S_t, A_t)$ 做迭代更新时，可以采用上一次迭代得到的 $q_\pi(S_{t+1}, A_{t+1})$。

Q 值矩阵的更新过程可以依据 Bellman Expectation Equation，其数学表示为：

$$q_\pi(s,a) = E_\pi[R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a] = R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a \sum_{a' \in A} \pi(a' \mid s') q_\pi(s', a')$$

通常，可以将上式表述为一个逐步迭代学习的表达式：

$$q_\pi(s,a) = E_\pi[R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a] = q_\pi(s,a) + \alpha[R_s^a + \gamma \max q_\pi(s', a') - q_\pi(s,a)]$$

不难理解，上式即是依据概率对更新 Q 值期望的一个估计。理论上可以证明，依据 Bellman Expectation Equation 来对 Q 值矩阵进行更新，Q 值矩阵是以 γ 的速率逐步收敛的。

除了对 Q 值矩阵做更新以外，还需要对依据最新的 Q 值矩阵对策略 π 进行更新。通常可以采用贪婪的方法来进行更新，也即选择当下状态最大 Q 值对应的动作作为策略。策略更新与 Q 值矩阵更新交替进行，直至迭代终点，即可求解此马尔科夫决策过程。

具体的 Q-Learning 算法流程如下所示：

Function Q-value Iteration

Initialize corresponding parameters and estimate model $\hat{P}$

For j=0,1,…,k-1 **do**

For each $s \in S$ **do**

$\pi_j(s) = \arg\max_{a \in A} Q_j(s,a)$ *# select greedy policy based on Q-value after iteration j*

For each $a \in A$ **do**

$$Q_{j+1}(s,a) = Q_j(s,a) + \alpha[R_s^a + \gamma \max Q_j(s',a') - Q_j(s,a)] \quad \# \ update \ \ Q\text{-}value \ Matrix$$

End for

End for

End for

End for

Return Q_k

4 辅助驾驶决策结果分析

4.1 辅助驾驶决策结果

辅助驾驶决策的结果，如图 14 所示。

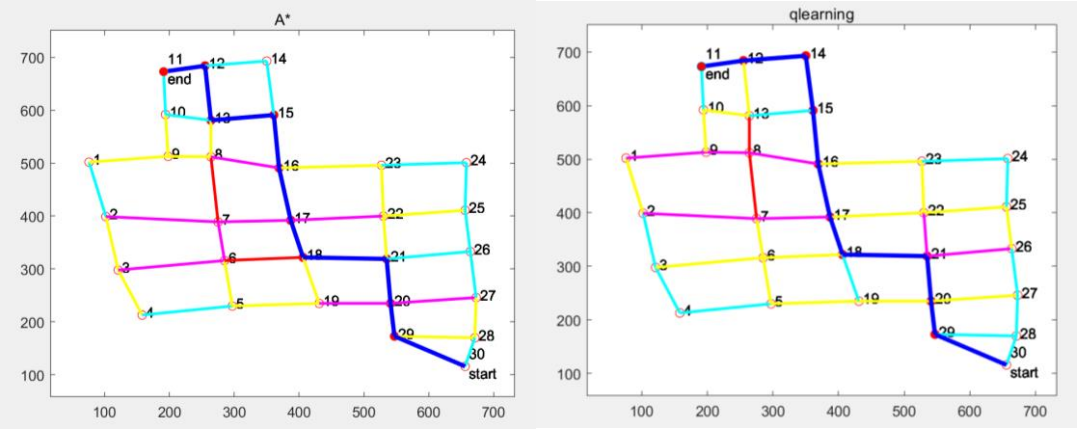

图 14 辅助驾驶决策结果分析

（左图为 A*的决策结果，右图为 Q-Learning 的决策结果）

根据上述结果，可以看出，A*和 Q-Learning 的辅助驾驶决策结果基本一致。其中，蓝色通路代表辅助驾驶决策生成的建议路线，每个路段的颜色代表其拥堵程度。由于实际程序为状态切换的动态过程，这里无法具体描述，详见附件的代码部分。

4.2 算法优缺点分析

4.2.1 A*算法的优缺点分析

- 由于 $h(n)$ 为启发式函数，所以 A*算法得到的结果可能不完全准确，存在随机性。
- 在高复杂度地图情况下，A*算法遍历搜索的代价较大，计算复杂度较高。
- A*算法在运行时需要不断扩充并迭代 openlist，在地图结点数目较多，结构较为复杂时，上述任务的时间复杂度会极大，因此 A*解决复杂地图规划问题的能力较弱。
- A*使用启发式路径规划算法来寻找最优路径，但是启发式算法导致 openlist 中存在

多个最小值时，A*可能无法选出最优路径。也就是说，启发式函数的存在导致 A*可能错失最优路径。

4.2.2 Q-Learning 算法的优缺点分析

3 Strengths & Weaknesses（QL）

（1）Q-Learning 算法的优点

- Q-Learning 实现了基于局部信息的辅助驾驶决策，现实情况下很多时候是无法获取到全局信息的，故而将 Q-Learning 算法用于辅助驾驶有非常实际的效果。
- Q-Learning 的 Q 值是当下 Reward 与未来 Reward 的加权和，将未来的回报进行了考虑。这一点是 A*算法所不具备的，故而理论上 Q-Learning 算法能够做出更为“远视”的决策。
- 在算法实现过程中，Q 值矩阵总是基于其前一状态的最优值来进行更新的。对于交通流这种短期内变化不会太剧烈的问题，相邻两个状态对应的 Q 值矩阵变化不大，故而在算法后期的 Q 值矩阵迭代更新会较快地收敛到真值。

（2）Q-Learning 算法的缺点

- Q-Learning 适用于解决小型的强化学习算法，因为在 Q 值矩阵迭代收敛过程中那个需要大量的遍历运算，其算法复杂度将随着状态-动作对 $\langle s,a\rangle$ 的增大而迅速增加。在问题求解过程中，我们也可以看到 Q-Learning 算法的效率是显著低于 A*算法的。故而，在解决大规模强化学习问题中，更多地采用基于深度学习的强化学习方法，能够显著降低算法复杂度。
- 在问题求解过程中，由于 Q 值矩阵需要长时间的迭代过程才能收敛，故而我们为其迭代设置了迭代上限。但是这一做法可能导致算法初期 Q 值矩阵距离真值较远，从而制定出较差的策略，导致算法初期可能会有一定的振荡。
- Q-Learning 中关于 Exploitation 与 Exploration 之间的权衡是较为主观的，过多地进行 Exploitation 可能导致收敛到局部最优解；而过多地进行 Exploration 又有可能导致 agent 较少地利用已有知识，从而做出较差的决策。

5 结论与展望

5.1 对于离散化处理的优缺点

本文在处理动态交通流问题时，大量使用了离散化建模的思想，构建了离散地图，并确定了离散的状态空间，即我们处理的车辆导航问题，其时间和空间都是高度离散化的，因此得以使用上述算法解决。离散化建模是交通建模中最常见的一种方法，包括地铁站点等交通系统都使用离散建模。

在交通流问题中，离散化处理可以大大简化问题，如在我们的案例中，车辆在一条道路

上如何变道，如何超车，以何种速度曲线行进等等问题并不是我们所要考虑的核心问题，也不显著影响司机的路线选择，而离散化建模忽略了这些问题，以有代价的边代替，使模型其可以被计算机程序求解；同时，离散化的模型允许计算时使用迭代式的计算方法，让动态的地图分析和计算成为可能。

另一方面，出于简化的思想，离散化必然导致部分信息被忽视。模型往往假设被忽视的信息与模型要解决的问题之间无关联，但当上述假设错误时，即忽略的信息和模型目标之间存在某种关联时，离散化建模容易导致应用效果较差。因此，离散化建模时的核心假设决定了其最终的执行效果。

5.2 基于伪全局信息的路径规划目标分解

无论是 A*还是强化学习等算法，适当地使用全局信息总能提高模型的运行结果，但是全局信息的引入又必然加大计算复杂度，因此，若辅助驾驶决策时，起点和终点的距离过大，之间的地图过于复杂，可以假设一定区域内的交通流环境与其他区域无关，使用目标分解的方式构建伪全局信息。

以浙江大学紫金港校区到杭州东站的辅助驾驶决策为例，这条线路距离长，且其横跨了杭州市的东西两个方向，地图较为复杂。在实际驾驶决策中，可以在线路上选定一到两个子目标，如图 15 中选择了德胜立交为子目标。

图 15 子目标选择示意图

选择子目标后，以紫金港校区为起点，德胜立交为终点进行辅助驾驶决策，使用这两点间的区域信息作为全局信息，并认为其他区域不影响该区域的交通流环境，完成路线规划任务。同理，到达德胜公交后，再以德胜公交和杭州东站为起终点进行决策。

子目标方法允许我们改变我们的区域信息窗口大小，能够有效地提高算法的运行效率，更快地作出反应，并且让我们的模型能够更好地适应于长距离的辅助驾驶决策任务。

参考文献

[1] 惠飞, 穆柯楠, 赵祥模. 基于动态概率网格和贝叶斯决策网络的车辆变道辅助驾驶决策方法[J]. 交通运输工程学报, 2018, v.18；No.92(02):152-162.

[2] Shur J K, Jung H G. Sensor fusion-based vacant parking Slot detection and tracking[J]. IEEE Transactions on Intelligent Transportation Systems, 2014, 15(1): 21-36.

[3] Mosquet X, Andersen M, Arora A. A roadmap to Safer driving through advanced driver assistance systems[R]. Washington DC: MEMA, 2015.

[4] Velez G, Otaegui O. Embedding vision-based advanced Driver assistance systems: a survey[J]. IET Intelligent Transport Systems, 2017, 11(3): 103-112.

[5] 张书玮, 罗禹贡, 李克强. 动态交通环境下的纯电动车辆多目标出行规划[J]. 清华大学学报(自然科学版), 2016(2):130-136.

[6] Schneider M, Stenger A, Goeke D. The electric Vehicle-routing problem with time windows and recharging Stations [J]. Transportation Science, 2013, 48(4): 500-520.

[7] Erogan S, Miller-Hooks E. A green vehicle routing problem [J]. Transportation Research, 2012, 48(1): 100-114.

[8] Ombuki B, Ross B J, Hanshar F. Multi objective genetic Algorithm for vehicle routing problem with time windows [J]. Applied Intelligence, 2006, 24(1): 17-30.

[9] Kobayashi Y, Kiyama N, Aoshima H, et al. A route search Method for electric vehicles in consideration of range and Locations charging stations[C]//2011IEEE Intelligent Vehicles Symposium. Baden-Baden, 2011: 920-925.

[10] Mariyasagayam M N, Kobayashi Y. Electric vehicle route Assistance using forecast on charging station[C]//ENERGY 2013: The Third International Conference on Smart Grids, Green Communications and IT Energy-Aware Technologies. Lisbon: IARIA, 2013: 134-142.

[11] Sachenbacher M, Leucker M, Artmeier A, et al. Efficient Energy-optimal routing for electric vehicles[C]//Proc 25th Assoc Advancement Artificial Intell Conf. San Francisco, CA, 2011: 1402-1407.

[12] 耿新力. 城区不确定环境下无人驾驶车辆行为决策方法研究[D]. 2017.

[13] Hart P E , Nilsson N J , Raphael B . A Formal Basis for the Heuristic Determination of Minimum Cost Paths[J]. IEEE Transactions on Systems Science and Cybernetics, 1968, 4(2):100-107.

附录

（1）团队成员风采照

（2）团队分工情况

姓名	工作量（%）
陈焱	16
吴诗琪	16
林润泽	16
戴清阳	16
曾之宸	20
陶新渝	16