

清 华 大 学

# 综 合 论 文 训 练

题目：无人驾驶环境下交叉口通行策略优化及仿真研究

系 别：自动化

专 业：自动化

姓 名：刘柏

指导教师：胡坚明 副教授

2017 年 5 月 31 日



# 关于学位论文使用授权的说明

本人完全了解清华大学有关保留、使用学位论文的规定，即：学校有权保留学位论文的复印件，允许该论文被查阅和借阅；学校可以公布该论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存该论文。

(涉密的学位论文在解密后应遵守此规定)

签 名：\_\_\_\_\_导师签名：\_\_\_\_\_日 期：\_\_\_\_\_



## 中文摘要

本项目以无人车与普通车混行的场景为研究对象，以在保证安全行驶的前提下最小化车辆通过交叉口的平均时间为研究目标，自主搭建了基于 MATLAB 的仿真平台，并设计了针对车辆转弯轨迹、多车协同策略和信号灯配时方案的优化方法。

在仿真平台搭建方面，本项目以典型的平面十字交叉路口为对象，完全自主设计并实现了能在给定车辆轨迹、协同策略、信号灯配时下进行车辆运动情况仿真的软件平台。在车辆转弯轨迹优化方面，本项目以平衡通行速度与乘客舒适度为目标，采用 Q-学习算法对无人车左转弯轨迹进行优化，并验证了其有效性与可行性。在多车协同策略优化方面，本项目将车辆两两编队，以之为整体应用 Q-学习算法进行协同策略优化，并在多种典型场景下验证了策略的有效性。在信号灯配时方案优化方面，本项目对多种智能优化算法的优化结果与运算性能进行分析对比，发现粒子群算法对本问题具有较好的适用性。

**关键词：**无人驾驶；交叉口通行策略；增强学习；智能优化算法

## ABSTRACT

The project focuses on the scene of self-driving vehicle and ordinary vehicle driving in a mixed way. Aiming at minimizing the average intersection travel time with safety guarantee, we independently build a simulation platform based on MATLAB, and design optimization method of vehicle turning trajectory, multi-vehicle coordination strategy as well as traffic signal timing scheme.

In the aspect of simulation platform implementation, we develop a software platform which can simulate the vehicle movement under given turning trajectory, coordination strategy and traffic signal timing scheme. Regarding vehicle turning trajectory, we optimize it using Q-learning algorithm to find the best tradeoff between speed and comfort, and verify its effectiveness and feasibility. When optimizing multi-vehicle coordination strategy, we pair vehicles and apply Q-learning algorithm for optimization, and verify the effectiveness of the strategy under typical scenarios. In the aspect of traffic signal timing scheme optimization, we analyze and compare the results and performance of several intelligent optimization algorithms, and find that the particle swarm algorithm stands out.

**Keywords:** Self-driving; Intersection travel strategy; Reinforcement learning; Intelligent optimization algorithm

# 目 录

第 1 章 引言 .....	1
1.1 选题背景及意义 .....	1
1.2 文献调研 .....	1
1.2.1 信号灯控制策略优化 .....	2
1.2.2 交叉口车流分导 .....	3
1.2.3 增强学习 .....	3
1.3 研究方案 .....	5
1.3.1 问题定义 .....	5
1.3.2 整体构思 .....	6
1.3.3 实现平台 .....	6
1.4 论文结构 .....	7
1.5 本章小结 .....	7
第 2 章 系统设计 .....	8
2.1 车辆 .....	8
2.2 路口 .....	9
2.2.1 拓扑结构 .....	9
2.2.2 通行规则设计 .....	11
2.3 信号灯 .....	11
2.4 行驶轨迹 .....	12
2.4.1 轨迹方案示意 .....	12
2.4.2 左转弯轨迹 .....	13
2.4.3 直行轨迹 .....	13
2.4.4 右转弯轨迹 .....	13
2.5 本章小结 .....	14
第 3 章 转弯轨迹优化 .....	15
3.1 原理概述 .....	15

3.2 参数设定 .....	17
3.2.1 学习速率因子与奖励折现因子 .....	17
3.2.2 训练次数 .....	18
3.2.3 贪婪参数 .....	18
3.3 状态空间 .....	19
3.3.1 参数选取 .....	19
3.3.2 精度选择 .....	20
3.3.3 状态空间简化 .....	21
3.4 策略集 .....	24
3.5 奖赏函数 .....	26
3.5.1 距离因素 .....	26
3.5.2 运动方向因素 .....	27
3.5.3 舒适度 .....	27
3.5.4 参数设置 .....	28
3.6 优化结果 .....	29
3.6.1 仿真轨迹 .....	29
3.6.2 最优轨迹设计 .....	30
3.6.3 轨迹有效性验证 .....	31
3.7 性能分析 .....	33
3.7.1 覆盖率 .....	33
3.7.2 训练耗时 .....	34
3.8 本章小结 .....	35
<b>第 4 章 多车协同策略优化 .....</b>	<b>36</b>
4.1 原理概述 .....	36
4.2 参数设定 .....	37
4.2.1 学习速率因子与奖励折现因子 .....	37
4.2.2 训练次数 .....	37
4.2.3 贪婪参数 .....	38
4.2.4 状态转移时间 .....	39
4.3 状态空间 .....	40



4.3.1 参数选取.....	40
4.3.2 范围选择 .....	41
4.3.3 精度选择.....	42
4.4 策略集 .....	43
4.4.1 无人车+无人车 .....	43
4.4.2 无人车+普通车 .....	44
4.5 奖赏函数 .....	44
4.5.1 速度因素.....	45
4.5.2 距离因素 .....	46
4.5.3 函数形式.....	47
4.6 优化结果 .....	48
4.6.1 无人车+无人车 .....	48
4.6.2 无人车+普通车 .....	52
4.7 性能分析 .....	55
4.7.1 覆盖率 .....	55
4.7.2 训练耗时.....	57
4.8 本章小结 .....	59
<b>第 5 章 信号灯控制策略优化 .....</b>	<b>61</b>
5.1 模型设计 .....	61
5.1.1 问题构建.....	61
5.1.2 优化方案.....	62
5.2 模拟退火算法 .....	62
5.2.1 原理 .....	62
5.2.2 参数设定.....	64
5.2.3 优化结果.....	65
5.2.4 性能分析.....	66
5.3 遗传算法 .....	67
5.3.1 原理 .....	68
5.3.2 算法应用.....	69
5.3.3 参数设定.....	71

5.3.4 优化结果.....	73
5.3.5 性能分析.....	73
5.4 粒子群算法 .....	74
5.4.1 原理 .....	75
5.4.2 参数设定.....	76
5.4.3 优化结果.....	77
5.4.4 性能分析.....	78
5.5 算法对比 .....	79
5.6 本章小结 .....	81
<b>第 6 章 总结和展望 .....</b>	<b>83</b>
6.1 总结 .....	83
6.2 展望 .....	83
插图索引 .....	85
表格索引 .....	87
参考文献 .....	88
致    谢 .....	91
声    明 .....	93
附录 A 外文资料的书面翻译 .....	95

# 第1章 引言

## 1.1 选题背景及意义

长久以来，无人驾驶一直是机械、控制、交通领域的热门话题。相比传统车辆，自动驾驶车辆具有可控性强、易于调度、智能性高、人力成本低廉等优点，在现代大规模、复杂交通系统中拥有十分明显的优势。而当下，中国对交通等基础设施建设设施的需求量较大，相关产业发展前景极为广阔。然而，由于种种原因，中国的大、中、小型城市均存在不同程度的交通问题，其中以北京、上海、广州、深圳为代表的特大型城市的交通堵塞问题格外严重。而无人驾驶车辆由于其可被集中调度的特性，在维持交通秩序、避免交通拥堵方面具有得天独厚的优势。事实上，调查显示，相较其他国家而言，中国对于缓解交通堵塞的需求更为迫切，而对无人驾驶的态度也更为积极<sup>[1]</sup>。此外，相关研究也显示，无人驾驶车辆在当代城市中具有巨大的潜力与影响，相关产业未来前景十分光明<sup>[2]</sup>。

当然，无人驾驶属于新兴研究领域，当前在技术上依旧存在着许多待解决的技术问题，但同时也意味着其具有十分巨大的研究潜力。例如，当前的无人驾驶导航技术即使是在较为简单的交通系统中也会产生极为惊人的计算量，而许多已有的系统也只能适用于较为有限的场景<sup>[3]</sup>。此外，由于法律尚待健全<sup>[4]</sup>等种种因素，当前对无人车的安全性能要求极为严苛。

而另一方面，交叉口是城市交通中不可忽视的一个环节。城市交通网络中，交叉口区域的等待时间与拥堵现象往往会在总旅行时间中占据非常明显的部分，例如在哥本哈根，交叉口区域的耗时在总行驶时间中所占据的比例可高达 50%<sup>[10]</sup>。但目前，经过广泛的文献调研，我们发现，关于无人驾驶车辆在路口的行驶策略的研究几乎是一片空白。

综上所述，无人驾驶领域前景广阔、需求量大，同时在技术上依旧存在着巨大的研究空间，是极为有价值的研究课题。其中，关于路口通行策略的研究一方面具有极为重大的战略意义，另一方面在目前相关成果还较为匮乏，故本项研究将具有较为显著的意义与贡献。

## 1.2 文献调研

本研究的研究对象是交叉口。经过文献调研,在交叉口通行策略优化方面,目前已有的研究成果主要有两类:基于信号灯控制策略的优化和基于交叉口车流分导措施的优化。此外,由于在本项目中,我们将采用增强学习的方法进行策略优化,故在前期调研过程中我们也针对增强学习的研究现状进行了文献调研。

### 1.2.1 信号灯控制策略优化

信号灯控制策略优化领域的相关成果较多,且方法较为多样,不少成果是基于智能优化算法的,同时也有从交通结构出发,利用运筹学相关数学工具进行机理建模的研究;而在优化对象方面,既有研究一般车辆的,也有专门进行公共交通工具控制的研究。此外,相关研究也涵盖了各式交叉口,包括拥堵型交叉口、公交优先型交叉口等。

对于通用的交叉口,经过适当的假设与简化,有的研究从解析的角度出发,利用动态规划的方法,将路口的延误、排队纳入考虑,针对信号灯配时进行优化<sup>[6]</sup>。而在不进行简化的情况下,由于交通系统极为复杂,难以进行数学特性良好的机理建模,因此智能优化算法、神经网络等非确定性优化方法往往能发挥较为良好的效果。例如,有研究结合系统控制理论,利用模糊控制方法,再构建三层前馈神经网络进行优化,取得了良好的效果<sup>[7]</sup>。已有的研究中,遗传算法,作为经典的智能优化算法,被广泛地运用。有的研究针对过饱和的交通状况,利用遗传算法最小化平均等待时间,获得了比较良好的效果<sup>[8]</sup>。

而针对具有特殊结构的交叉口的研究也较为丰富。在有专门的右转车道的情况下,有研究基于 DCC (Degree of Clustered Conflict) 的方法,实现了信号灯的在线控制,不仅提升了路口通行效率,也降低了潜在的安全风险<sup>[5]</sup>。而在车路协同的场景下,有研究通过利用通信信息,消除了车辆在交叉口的潜在轨迹交叠,极大地提升了安全性<sup>[9]</sup>。

此外,关于公共交通工具在交叉口的通行研究也是一个热门领域。考虑到公共交通工具具有较高的荷载量且在缓解城市交通拥堵方面具有的重要作用,是否应在交叉口安排公共交通工具优先通行一直是一项热门的研究的话题,其对于城市交通政策也具有相当重要的意义。例如,通过电脑仿真的方法,有研究针对单个交叉口进行了不同方案的比较研究,认为公交优先通行在经济上是否更优取决于交通网络结构与交通流特征<sup>[11]</sup>。而近年来,在相关主题上,有研究利用了公共交通工具运行的规律性与可预测性,基于该部分信息,通过优化在一级、二级交

叉口的信号配时方案，提出了在允许公共交通工具优先通行的前提下的优化策略，并显著降低了衍生的不利影响<sup>[12]</sup>。

### 1.2.2 交叉口车流分导

相比通过改变信号配时的方式优化交叉口通行状况，对交叉口车流引导措施的研究则较为少见。

由于交叉口结构与车辆行为的复杂性，相关研究多采用仿真手段，而结果几乎都表明，合理的交通分导策略能改善交叉口的通行情况。例如，有研究设计了三种分导方案，在泰州市的背景下进行仿真，得到了较为理想的分导方案，而该方案也具有较为良好的普适性<sup>[13]</sup>。还有研究通过敏感性分析的方法，研究了在信号灯路口设置左转车流引流的效果<sup>[14]</sup>。

此外，也有研究将交叉口车流分导与信号灯配时策略优化相结合，提出了将信号灯配时与车流分导依次进行优化，再迭代多次得优化方案，通过仿真说明尤其适用于交通较为拥堵的交叉口<sup>[15]</sup>。

### 1.2.3 增强学习

增强学习是机器学习的一个分领域，在控制领域展现了极为出色的优化效果。增强学习在思想上主要借鉴了生物的行为模式：一个生物个体如何通过不断地尝试、接受反馈、作出调整，趋利避害进而获得最大的总奖励<sup>[16]</sup>。

增强学习具有较强的交叉性的特点，与许多其他学科，如控制论、博弈论、信息论、运筹学、仿真优化、集群优化、多目标优化、统计学、随机过程等具有密不可分的关系。而其应用领域也十分广泛，包括参数优化、运动控制优化、路径规划、博弈决策等。

增强学习方面的研究至今已经有 40 余年的历史<sup>[17]</sup>。最初的增强学习的应用场景是计算机科学，与自适应动态规划（ADP）和神经动态规划（NDP）较为相似。而到了 20 世纪 90 年代左右，增强学习在理论上有了新的突破<sup>[18, 19, 20, 21]</sup>。在 21 世纪初期，随着人工智能的兴起，不少增强学习领域的研究开始涉及脑科学，以期从增强学习的角度进一步理解人脑运作机制<sup>[22, 23, 24, 25]</sup>。而近些年来，通过观察增强学习在具体领域的应用，我们可以发现，增强学习具有极强的普适性，可被用于解决许多传统领域的问题。

在仓储管理问题方面，如何进行合理调度，进而减少零售商仓储开销一直是网络优化领域的一个热门问题。传统的动态规划方法一直是行之有效的手段，但

在有的应用场景中，由于模型的状态空间较为庞大，导致传统算法的计算复杂度非常高。针对这一情况，有研究采用了与增强学习较为类似的神经动态规划方法，将仓储开销降低了 10% 左右<sup>[26]</sup>。

在数据库系统方面，由于数据库在当今的作用越来越重要，且当今的数据越来越海量，人们对数据库的功能提出了更高的要求，其中一条就是要求数据库系统能描述非文本信息。关于非文本信息的描述，一种通行做法是采用元数据解释器。针对这一场景，为了提升数据库管理效率，有研究采用增强学习的方法以提升数据库操作的集成性，并方便用户与数据库之间的交互操作<sup>[27]</sup>。

在电力系统控制方面，如何根据获取的电网相关信息来对电网电力进行调度与调控一直是一项热门的研究主题。其中一类研究问题是：如何设计智能体，使其能够根据系统反馈的信息学习出较优的控制策略，进而部署到电网控制设备中。针对这一问题，有研究将其视为离散时间最优控制问题，将系统反馈的信息视为采样信号，通过增强学习的方法进行一步一步的值迭代，进而估算出系统的策略函数，获取最优控制策略<sup>[28]</sup>。

增强学习也适用于调度问题，例如航班流量控制问题。航空气调度系统具有自适应、可重构的特点，尤其适合使用增强学习的方法进行控制优化。针对航空气调度问题，有研究采用增强学习中的估计对象参数（ADP）方法，通过利用待优化对象与环境之间互动获得的信息，让其能够在线自适应地进行优化。该项研究在线下优化中，将估计对象参数方法与经典的 F-16 算法进行对比，发现估计对象参数方法的成功率显著高于 F-16 算法；而在线上优化中，该项研究将估计参数方法与控制依赖型算法进行对比，发现估计参数优化方法依旧具有更优的效果<sup>[29]</sup>。

而在动态能量管理系统中，增强学习也尤为适用。在智能电网中，能源管理系统的代价函数在结构上具有子模性和单调性的特点，使得我们能够通过采用增强学习中的改进 Q-学习的方法，在策略空间中取得较为理想的权衡结果，并收敛得到更优的理想的控制策略。有研究基于这一思想，提出了线性自适应算法，通过拉格朗日乘子搜寻使得信息传输能耗最小的能源管理策略，使得在保证较好的稳定性和鲁棒性的前提下，在虚拟和现实场景的工作环境下，能耗分别减少 30% 与 60% 之多<sup>[30]</sup>。

时下，增强学习在运动控制领域的应用十分流行，例如，有研究将基于增强学习的自适应动态规划方法（ADP）用于四旋翼直升机控制平台的优化。取得了较为良好的效果<sup>[31]</sup>；还有研究将自适应动态规划方法和无模型的 Q-学习算法用于机械臂的控制，均取得了令人较为满意的结果<sup>[32, 33]</sup>。不过，增强学习在实际的运

动控制中，依旧面临着比较严峻的问题，例如，在实际情况中，状态空间的规模往往极为庞大，增强学习的计算量也将十分庞大，使其难以胜任运动控制中对时效性的要求。

### 1.3 研究方案

本项研究基于无人驾驶的背景，旨在针对无人车与普通车在交叉口混合通行的策略进行优化。

#### 1.3.1 问题定义

根据 1.2.1 节和 1.2.2 节的文献综述，信号灯配时方案的设计是极为重要的交叉口通行策略优化方法。此外，由于我们所研究的对象是具有较强的可控性的无人车，我们得以针对无人车在交叉口的转弯轨迹进行优化。考虑到我们研究的场景是无人车与普通车混行的交叉口，车辆之间的协同控制策略也是优化对象。

我们以典型的双向多车道十字形平面交叉口为研究对象，包含小段车道、转角、转弯公共区域等。其平面示意图如图 1.1 所示。

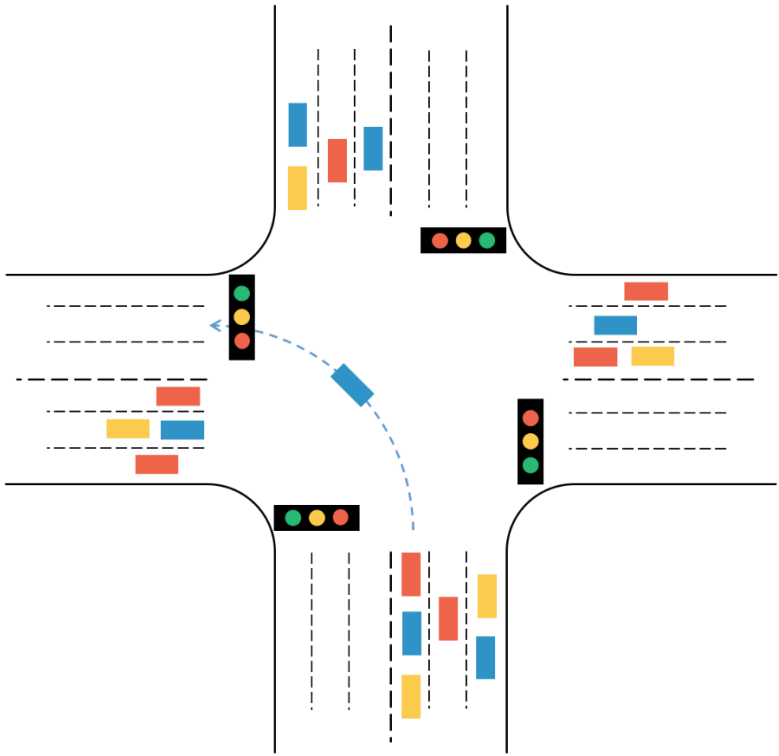


图 1.1 路口结构示意图

因此，本项研究所针对的问题定义为：在无人车与普通车混行的背景下，如何设计针对车辆转弯轨迹、多车协同策略和信号灯配时方案进行优化的策略，使得在保证安全的前提下，车辆通过交叉口的平均时间最少。

### 1.3.2 整体构思

我们要优化的对象是车辆转弯轨迹、多车协同策略和信号灯配时方案，三者可分别采用以下方法进行优化。

针对车辆转弯轨迹优化，一方面，这一过程较为偏向运动控制，且增强学习目前在运动控制领域展现了十分良好的控制效果，而另一方面，车辆的行驶过程可视为马尔科夫过程，故本项研究将采用增强学习的方法对交叉口车辆转弯策略进行优化。而在增强学习中，Q-学习是一种无模型的学习算法，尤为适用交叉口这种较为复杂的系统优化，故本项研究将采用 Q-学习算法。

针对多车协同策略优化，考虑到多车协同过程的复杂性，我们将继续采用无模型的增强学习方法。由于现有的较为成熟的多智能体增强学习算法所研究的多目标对象是时不变的。而在本问题中，行驶在特定的车道上的车辆是随时间变化的，难以直接应用传统的多智能体增强学习的方法。为此我们考虑将车辆进行分组，以一组车辆的整体为研究对象，使得该组内车辆各自的状态与它们之间的位置关系共同构成状态空间，如此即可将问题转化为单智能体优化问题，进而可以用 Q-学习算法进行求解。

针对信号灯配时方案优化，由于信号灯配时的机制极为复杂，但可优化参数较为有限，故可采用智能优化算法进行优化。在已有的智能优化算法中，模拟退火算法、遗传算法和粒子群算法具有较高的成熟性、较强的普适性，且实现方法较为简单、运算效率较高，故本项研究将仿真实现并对比这三种算法，从中选择最适合交叉口通行策略的算法。

### 1.3.3 实现平台

由于本项研究注重算法设计，且控制方案综合性较强，故选择采用自行编写仿真平台的方式。具体情况如下：

- (1) 语言：MATLAB;
- (2) 环境：MATLAB for Mac R2016b;
- (3) 环境：macOS Sierra 10.12.5。



## 1.4 论文结构

本篇论文结构如下：

第 1 章为引言，介绍本项目选题背景及其意义，并进行文献调研、制定研究方案。

第 2 章介绍了基于 MATLAB 的仿真系统在设计与实现方面的细节。

第 3 章介绍了利用 Q-学习算法进行转弯轨迹优化的方法，详细阐释了设计细节，并对仿真结果和性能进行分析。

第 4 章将 Q-学习算法应用在多车协同策略优化问题上，根据问题特性在理论层面对算法进行改进与设计，并对多种应用典型场景进行了仿真与测试。

第 5 章以优化信号灯配时方案为目标，分别采用模拟退火算法、遗传算法和粒子群算法进行优化并对各方案进行了综合对比。

第 6 章对全文的内容进行总结，并提出了今后在相关研究方面的可能拓展方向。

## 1.5 本章小结

在本章，我们先结合当前国内外的交通现状与无人驾驶技术的发展情况，阐述了本项目的选题背景及意义。

之后我们对国内外的相关研究进行了文献调研，发现目前几乎还没有针对无人车在交叉口通行方案的研究。

我们接着制定了研究方案，介绍了整体的研究思路。

我们最后阐述了各章节的结构。

## 第2章 系统设计

由于在本项目中，不论是 Q-学习还是智能优化算法，仿真系统都在其中起到了重要作用，故仿真平台的物理模型设计将具有极为重要的作用与意义。

在本项目中，基于 MATLAB 平台，我们自行设计并完全自主搭建了仿真系统。我们将从车辆、路口、信号灯和车辆轨迹的角度对仿真平台进行说明。

### 2.1 车辆

本项目所设计的车辆模型包含的参数有两类：静态属性（编号、尺寸、类型、既定线路）、动态属性（包含速度和加速度的动力指标、位置、当前状态等）。各参数的介绍如表 2-1 所示。

表 2-1 车辆主要参数

参数名称	描述	单位
编号	标识车辆的唯一编号	-
长度	车身长度	m
宽度	车身宽度	m
类型	车辆类型（无人车/普通车）	0/1
进入车道编号	车辆进入交叉口的车道的编号	-
目的车道编号	车辆离开交叉口的车道的编号	-
速度	车辆当前速度	$\text{m} \cdot \text{s}^{-1}$
加速度	车辆当前加速度	$\text{m} \cdot \text{s}^{-2}$
横坐标	车辆当前位置的横坐标	m
纵坐标	车辆当前位置的纵坐标	m
角度	车辆当前行驶的角度	rad
轨迹	车辆以往行驶过的位置的集合	-
状态	未进入/在交叉口/已驶出	0/1/-1

由于本项目的研究对象为无人车与普通车的混合通行模型，故需要通过“类型”属性将其区分。值得注意的是，除了“类型属性”之外，在其他属性方面，无人车与普通车辆并没有本质差别。

值得一提的是，我们专门为车辆增加了“轨迹”属性，用于记录该车在交叉口通行过程中每一时刻所处的位置，便于精细化分析车辆的运动状况以及绘制图像、动画等。

## 2.2 路口

路口部分主要由道路和路口的拓扑结构以及车道的通行规则所刻画。分别进行讨论如下。

### 2.2.1 拓扑结构

为了进行定量的建模与分析，我们将图 1.1 中的路口结构进行抽象化与量化，将其放置在平面直角坐标系中加以表示，如图 2.1 所示。

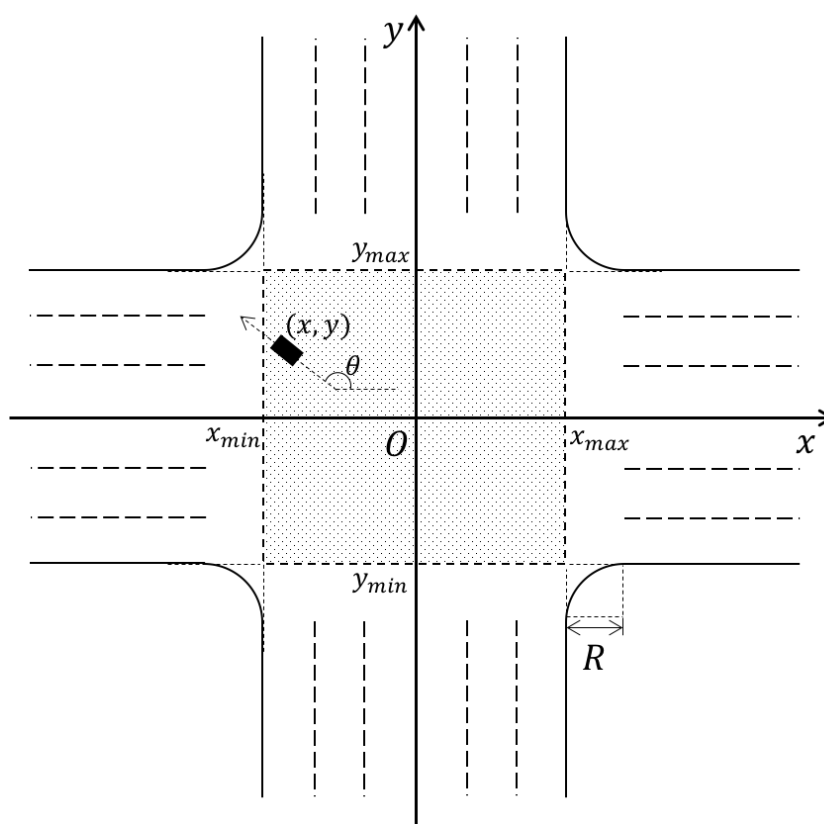


图 2.1 路口参数

我们研究的路口由东西走向和南北走向的双向车道正交组成，为此，我们可以很方便地建立平面直角坐标系：以东西走向的双向车道的分界线为  $x$  轴，以南北走向的双向车道的分界线为  $y$  轴，两轴交点为原点（也是交叉口的中心点）。如此一来，车辆的位置状态便可被清晰地描述。

我们以图中的黑色矩形示意车辆。矩形的中心坐标即为  $(x, y)$ ，而车辆前行方向与  $x$  轴正半轴所成的角度为  $\theta$ ，向量  $(x, y, \theta)$  即可描述完整描述车辆的位置状态。

将道路边缘延长，会在交叉口处构成矩形（如图 2.1 中阴影区域所示）。考虑到该部分区域覆盖了交叉口的大部分面积，且形状规则，便于进行数学建模与计算，我们将阴影部分的矩形区域作为进行轨迹优化时的定义域。

除此之外，为了使路口的抽象模型更为规则，从而方便建模并简化计算，我们依据实际情况进行设定如下：

- (1) 根据国家标准，标准公路的宽度为 3.75 m，故所有车道的宽度均相等；
- (2) 车道连接处的拐弯可用四分之一圆弧表示。

根据上述讨论，可对路口的各参数进行定义如表 2-2 所示。

表 2-2 路口结构主要参数

参数名称	描述	单位
编号	标识路口的唯一编号	-
长度	交叉口区域内车道的长度	m
车道数	单向车道数	-
车道宽度	单个车道的宽度	m

值得注意的是，上述参数不一定会在仿真过程中被全部用到，但仍予以保留，以加强可拓展性。例如，在本项目中，我们仅研究单个交叉口的通行策略问题，在仿真时并不会使用“编号”参数，但考虑到本系统可能作为路网的一部分，故在此处仍将其予以保留；此外，由于本项目主要研究的对象是阴影部分的矩形区域，故“长度”参数在仿真中也未起到作用。但考虑到，在未来可能会需要研究车辆在进入交叉口之前的道路部分的运动情况，故交叉口部分内部的车道长度应该予以保留。

### 2.2.2 通行规则设计

在进行仿真时，路口各车道的通行规则对于仿真过程至关重要。为了简化仿真过程，提升运算速率，我们对各类车辆的车道行驶作出如下规定（假设该方向上有  $n$  条车道，由外向内分别编号 1 至  $n$ ）：

- (1) 左转弯车辆仅能行驶在车道  $n$ ；
- (2) 直行车辆可行驶在车道 1 至车道  $n-1$ ；
- (3) 右转弯车辆仅能行驶在车道 1。

在这样的安排之下，在车道 1 上直行车辆和右转弯车辆混行，车道 2 至车道  $n$  上仅有直行车辆行驶，车道  $n$  上仅有左转弯车辆行驶。

在该套通行规则之下，再与 2.3 节的信号灯相位设计相配合，车辆轨迹相互不相交，路口行驶的车辆碰撞将可能仅仅来源于车辆的前后追尾，在安全性的角度极大地简化了仿真和优化过程。

## 2.3 信号灯

在本项研究中，我们采用固定周期的信号灯作为研究对象，即信号灯周期的时间长度一定，而在一个信号灯周期内信号灯的各个相位依次出现。在本项研究中，我们所研究的信号灯具有 4 种相位，如图 2.2 所示。

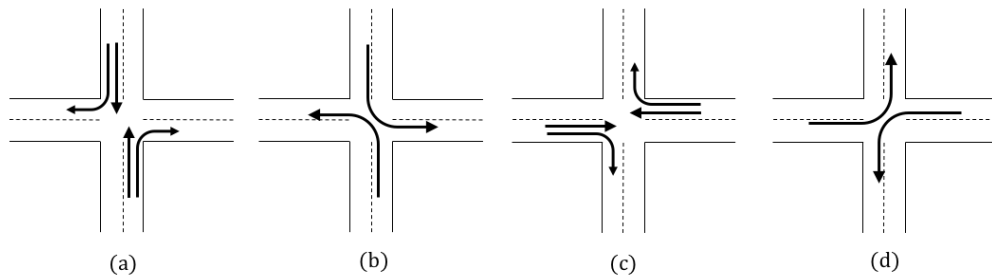


图 2.2 交叉口通行模式

在一个周期内，图 2.2 中的 4 种相位将依次出现。(a) 相位为南北走向的车辆进行直行和右转弯；(b) 相位为南北走向的车辆进行左转弯；(c) 相位为东西走向的车辆进行直行和右转弯；(d) 相位为东西走向的车辆进行左转弯。

如此设计的一个考虑是，在一般的十字路口中，拥堵情况发生的一个常见原因是允许直行与左转弯双向同时进行，导致直行车辆与左转弯车辆的行驶轨迹产

生交叉，容易因争抢优先行驶权而堵塞甚至导致事故，故我们将左转弯与直行的相位分开。

另一方面，右转弯若与左转弯处于同一相位，右转弯车辆与相向行驶的左转弯车辆可能出现并道问题，增加了仿真难度的同时也破坏了车辆协同策略的统一性。而若为右转弯单独设立相位，则交叉口通行效率会受降低。故最终，我们采取将右转车辆与直行车辆放置于同一相位的设计。

## 2.4 行驶轨迹

在仿真平台中，轨迹的设计会对仿真过程与仿真结果产生重要的影响，我们将在本小节详细介绍路口通行时的各类轨迹的设计方案。

### 2.4.1 轨迹方案示意

我们以在双向六车道平面十字路口中，从  $y$  轴负半轴向  $y$  轴正半轴方向行驶的车辆为对象进行说明。

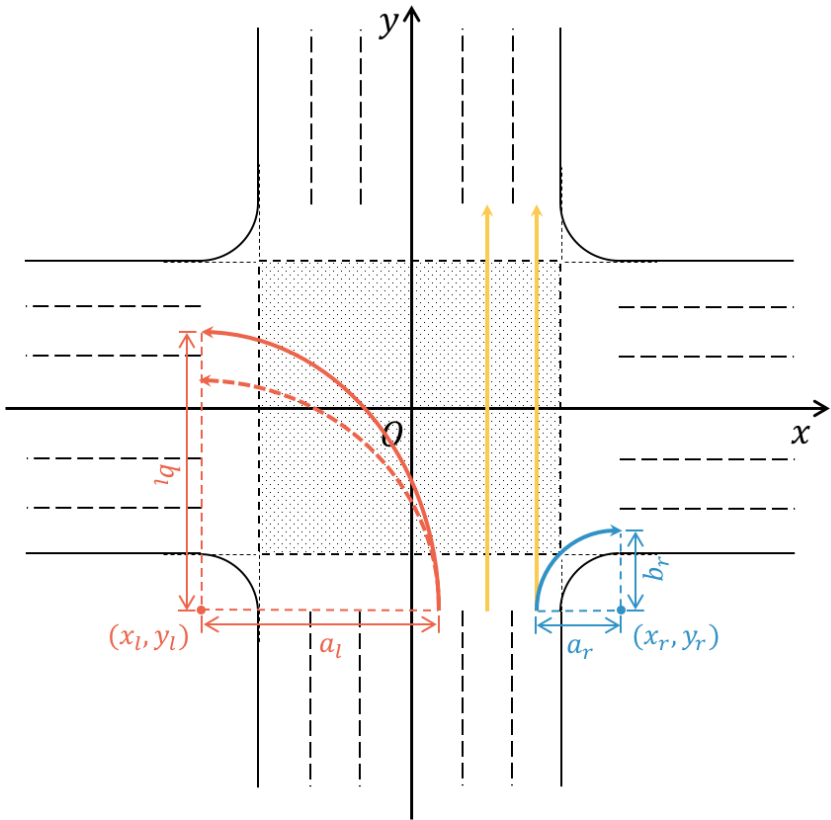


图 2.3 车辆行驶轨迹示意图

示意图如图 2.3 所示，我们将轨迹分为左转轨迹（红色）、直行轨迹（黄色）和右转轨迹（蓝色）。我们将接下来介绍各轨迹的详细设计方案。

### 2.4.2 左转弯轨迹

左转弯轨迹为图 2.3 中的红色轨迹。

由于我们将对左转轨迹中的进行优化，故在设计左转轨迹时，我们将分无人车轨迹与普通车轨迹讨论。

对于无人车，其行驶轨迹是经过增强学习算法优化的轨迹，具体方法以及结果将在第 3 章中详细讨论与说明，在此处用红色虚线表示。

而对于普通车，根据平时观察，普通车辆的转弯轨迹可用圆弧近似，在此处用红色实线表示。在本例中，我们用椭圆轨迹的四分之一弧描述车辆左转弯的轨迹：

$$\frac{(x - x_l)^2}{a_l^2} + \frac{(y - y_l)^2}{b_l^2} = 1 \quad (x \geq x_l, y \geq y_l) \quad (2-1)$$

其中各参数的物理含义参见图 2.3。

### 2.4.3 直行轨迹

直行轨迹为图 2.3 中的黄色轨迹。

由于我们研究的交叉口较为规整，直行车辆只需沿直线行驶即可完成交叉口的通行过程。故我们用直线描述直行轨迹。

### 2.4.4 右转弯轨迹

右转弯轨迹为图 2.3 中的蓝色轨迹。

在研究右转弯轨迹时，考虑到右转弯用时较短，故我们在设计右转弯轨迹时，不再将无人车与普通车分别对待。

而根据平时观察，普通车辆的转弯轨迹可用圆弧近似。在本例中，我们用椭圆轨迹的四分之一弧描述无人车与普通车右转弯的轨迹：

$$\frac{(x - x_r)^2}{a_r^2} + \frac{(y - y_r)^2}{b_r^2} = 1 \quad (x \leq x_r, y \geq y_r) \quad (2-2)$$

其中各参数的物理含义参见图 2.3。

## 2.5 本章小结

在本章，我们从车辆设计、路口设计、信号灯设计与轨迹设计四方面系统地介绍了我们所搭建的仿真系统。

在车辆设计方面，我们从车辆的静态属性（编号、尺寸、类型、既定线路）与动态属性（包含速度和加速度的动力指标、位置、当前状态等）介绍了车辆的属性。

在路口设计方面，我们一方面通过示意图严格建立了路口的抽象数学模型，并详细描述了路口各参数，另一方面根据现有交通通行规则，结合安全通行的需求，设计并详细论证了一套交叉口通行规则。

在信号灯设计方面，我们介绍了信号灯的相位组成情况以及各相位下的交通通行规则。我们还与路口设计中的交叉口通行规则相结合，论证了各相位通行规则在安全通行方面的有效性。

在行驶轨迹设计方面，我们绘制了轨迹示意图，并建立了左转弯、直行和右转弯三种情况下的轨迹方程。



## 第3章 转弯轨迹优化

由于直接对二维平面内的多智能体利用增强学习进行优化将会需要非常庞大的状态空间，导致计算效率低下甚至在现有硬件条件下无法计算，故我们先以单个智能车为研究对象，利用 Q-学习算法对车辆左拐时的转弯轨迹进行优化。

在本章节，我们将先介绍 Q-学习算法的原理，接着对 Q-学习算法在本问题场景下的部署与应用进行详尽的解释与分析，最后给出包括训练效果、训练性能等指标的仿真结果。

### 3.1 原理概述

增强学习在思想上主要借鉴了生物的行为模式：一个生物个体如何通过不断地尝试、接受反馈、作出调整，以趋利避害为指导思想，最终获得最大的总奖励。由于交通系统较为复杂，难以进行机理建模，故我们采用不依赖具体模型的 Q-学习算法。

在一般的贪婪算法中，智能体在每一步都会选择即时回报值最大的行动，但如此一来很容易陷入局部最优。为了避免此种情况发生，我们希望将选取某一行行动之后在未来获得的总回报也纳入决策信息，为此，增强学习的概念应运而生。

在增强学习中，我们的训练对象是  $Q(s_t, a)$ ，即 Q 函数。其中， $Q(s_t, a)$  表示在  $t$  时刻，在考虑未来获得的总回报的情况下，在状态  $s_t$  采取行动  $a$  所带来的回报值。Q-学习算法的目的正在于求解 Q 函数。

Q-学习算法的大致步骤是：

- (1) 设定学习速率因子  $\alpha_t$ ，奖励折现因子  $\gamma$ ，训练次数  $N$ ；
- (2) 令  $Q \leftarrow 0$ ；
- (3) 随机初始化当前状态  $s_t$ ；
- (4) 根据  $\epsilon$ -贪婪方法选择行动  $a_t$ ：

$$a_t = \begin{cases} \max_a Q(s_t, a), & \epsilon \in (rand, 1] \\ \text{随机选择 } a \in \{a_t\}, & \epsilon \in [0, rand] \end{cases} \quad (3-1)$$

- (5) 计算奖赏值  $r_{t+1}$ ；

(6) 更新  $Q(s_t, a_t)$  如下：

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha_t \left( r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t) \right) \quad (3-2)$$

(7) 更新当前状态：  $s_t \leftarrow s_{t+1}$

(8) 判断是否进入吸收态，若是，进入步骤 (9)，若否，返回步骤 (4)；

(9) 判断是否达到循环次数  $N$ ，若是，进入步骤 (10)，若否，返回步骤 (3)；

(10) 算法停止，输出结果。

其流程图如图 3.1 所示。

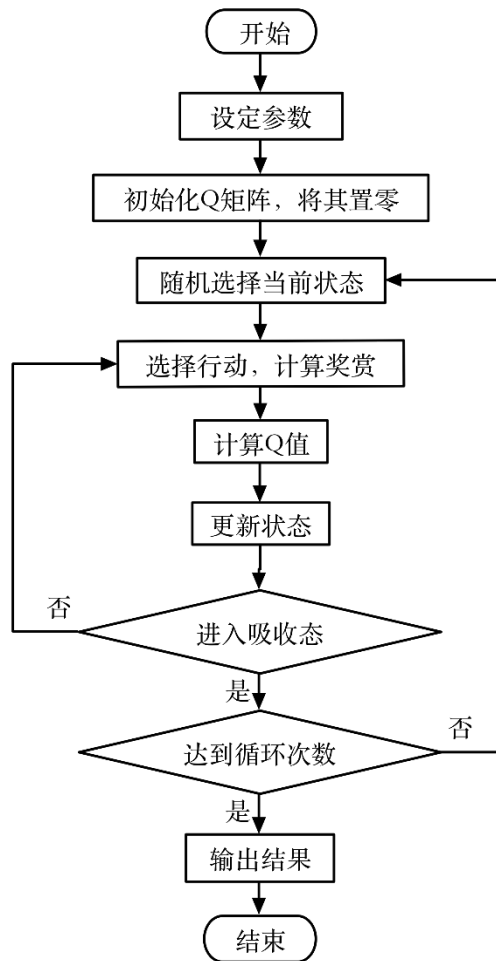


图 3.1 增强学习算法流程图

在获得 Q 函数之后，无人驾驶车辆只需根据当前状态，按照如下准则选取行动

$$a_t = \arg \max_a Q(s_{t+1}, a) \quad (3-3)$$

利用马尔科夫性和贝尔曼最优方程，可以证明，若 Q-学习算法满足以下条件，最终将能收敛得到全局最优：

- (1) 整个系统可被视为确定性的马尔可夫过程；
- (2) 即时奖赏值是有界的；
- (3) 智能体能无限次地遍历可行状态空间。

本项目按照以上准则设计了 Q-学习算法，在理论层面获得了全局最优的保证。

### 3.2 参数设定

Q-学习算法中的参数设定主要包括学习速率因子  $\alpha_t$ ，奖励折现因子  $\gamma$ ，训练次数  $N$  和  $\epsilon$ -贪婪方法中的  $\epsilon$  参数。

#### 3.2.1 学习速率因子与奖励折现因子

将 Q 函数更新公式(3-2)等价改写为

$$Q(s_t, a_t) \leftarrow (1 - \alpha_t) \cdot Q(s_t, a_t) + \alpha_t \gamma \max_a Q(s_{t+1}, a) + \alpha_t r_{t+1} \quad (3-4)$$

再假设  $Q(s_t, a) \approx Q(s_{t+1}, a) \approx Q$ ， $r_{t+1} \approx r$  以及  $\alpha_{t+1} \approx \alpha$ ，可将  $Q(s_t, a_t)$  的更新值近似为

$$(1 + \alpha\gamma - \alpha) \cdot Q + \alpha r \quad (3-5)$$

分析更新公式，考虑到  $Q$  本身是多次训练的结果，包含的信息量大于即时奖赏值  $r$ ，故  $Q$  的权重应显著大于  $r$ 。可构建约束条件如下：

$$\begin{cases} 1 + \alpha\gamma - \alpha > k\alpha \\ k > 1 \end{cases} \quad (3-6)$$

经过反复调试，最终发现按照式(3-7)设定参数能取得较为不错的效果。

$$\begin{cases} \alpha = 0.5 \\ \gamma = 0.5 \\ k = 1.5 \end{cases} \quad (3-7)$$

### 3.2.2 训练次数

理论上，只要训练次数足够多，所有状态都能得到遍历，且 Q 函数也将收敛到全局最优解。但在本问题中，状态空间的大小高达百万（参见 3.3 节），几乎无法通过增加循环次数使得 Q 函数收敛到最优解。因此，权衡计算量与训练结果，进行循环次数的设定至关重要。

在设定循环次数  $N$  时，我们主要参考了训练次数与状态空间覆盖率之间的关系（参见 3.7.1 节），最终发现  $N = 10000$  能在计算量与效果之间取得较好的平衡。

### 3.2.3 贪婪参数

Q-学习算法的训练效果对  $\epsilon$ -贪婪方法中的  $\epsilon$  参数高度敏感： $\epsilon$  较大时，智能体更倾向于随机选择下一步行动，以完成对未知状态空间的探索； $\epsilon$  较小时，智能体则更倾向于根据 Q 函数选择 Q 值最大的下一步行动，以进一步加强已知的较优策略。

考虑问题本身的性质，我们发现  $\epsilon$  参数的选取也与训练的进程阶段密切相关，训练开始与结束的场景对于状态空间的探索需求也并不一样。

在训练开始时，由于状态空间只有极小一部分得到了探索，Q 函数的大部分值为 0，而已经训练过的 Q 值可能对应的策略是局部较优甚至较差的，如果根据 Q 函数选择 Q 值最大的下一步行动，则很有可能使得非最优策略得到强化，训练效果不佳。故在此时， $\epsilon$  参数应该较大，以帮助智能体尽可能多地探索初始状态空间。

在训练开始之后，随着训练次数的增加，状态空间不断被探索，最优策略也逐渐得到强化。此时，如果  $\epsilon$  参数较大，智能体将继续探索非最优空间，在一定程度上会降低算法运行效率，且也使训练结果更容易陷入局部最优。

考虑到训练的不同阶段对  $\epsilon$  参数的不同需求，我们将折扣因子引入  $\epsilon$ -贪婪方法，在初始时设定较大的  $\epsilon$  参数，而每循环一次都将  $\epsilon$  以一定的比例进行折扣，具体而言，

$$\epsilon_{n+1} = \delta \cdot \epsilon_n \quad (0 < \delta < 1) \quad (3-8)$$

经过反复试验，发现在  $N = 10000$  时，选取

$$\begin{cases} \epsilon_0 = 0.2 \\ \delta = 0.999 \end{cases} \quad (3-9)$$

能取得较好的效果。

### 3.3 状态空间

在增强学习中，状态空间的设计起到了至关重要的作用。在设计状态空间时，需要根据问题的性质对构成状态空间的参数进行选取。而在 Q-学习算法中，状态空间需要进行离散化处理，而在离散化过程中，在精度与计算量之间进行权衡取舍也关键的环节。此外，考虑到本问题的应用场景中，状态空间较为庞大，我们还采取了为 Q 函数部分可行域赋无穷小初始值的方法减少了实际可行状态空间的大小，提升了运行速度。

#### 3.3.1 参数选取

在状态空间的参数对象选取方面，考虑到我们研究的目标是让单个智能体学习出最优轨迹，故状态空间应该由与车辆运动情况相关的动态参数组成。参数选取结果如表 3-1 所示。

表 3-1 状态空间参数选择		
名称	符号	单位
横坐标	$x$	m
纵坐标	$y$	m
速度方向	$\theta$	°

表 3-1 中，各项参数对应的物理含义如图 3.2 所示。

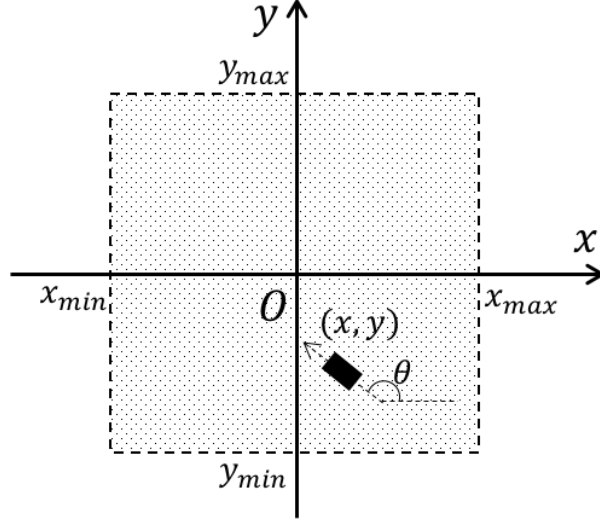


图 3.2 状态空间物理含义

可见，我们只选取了最为基本与重要的位置、方向信息，而没有考虑速度、加速度等动力学信息。如此进行参数选取的主要原因在于，车辆在下一时刻的可能位置是由速度决定，故状态空间转移过程能反映出车辆的速度信息；此外，在搜索下一时刻的状态空间可行域时，加速度的存在会使得车辆行驶速度发生变化，故加速度信息可由每一时刻的状态空间可行域体现。

### 3.3.2 精度选择

由于 Q-学习算法需要离散的状态空间进行训练，故需要将各参数进行离散化处理。

假设车辆在横坐标、纵坐标和速度方向的精度分别为  $\Delta x$ ， $\Delta y$ ， $\Delta \theta$ ，交叉口轨迹在横向与纵向的大小分别为  $X$ ， $Y$ ，则状态空间的大小为

$$|S| = \frac{X}{\Delta x} \cdot \frac{Y}{\Delta y} \cdot \frac{360}{\Delta \theta} = \frac{360XY}{\Delta x \Delta y \Delta \theta} \quad (3-10)$$

在双向四车道的情境中，查阅国家相关标准可知，标准公路的宽度为 3.75 m，故在该情境下状态空间大小为

$$|S| = \frac{360XY}{\Delta x \Delta y \Delta \theta} = \frac{360 \times (4 \times 3.75)^2}{\Delta x \Delta y \Delta \theta} = \frac{81000}{\Delta x \Delta y \Delta \theta} \quad (3-11)$$

在实际测试中发现，综合考虑计算机的运算能力以及在  $N = 10000$  时的收敛效果，状态空间不宜太大，应满足

$$|S| = \frac{81000}{\Delta x \Delta y \Delta \theta} < 2000000 \quad (3-12)$$

由于在进行轨迹优化时，车辆的速度方向信息相对位置信息而言明显更为不敏感，而轨迹在横向与纵向具有对称性，故可引入约束

$$\Delta \theta > 10 \Delta x = 10 \Delta y \quad (3-13)$$

此时，关系式可改写为

$$\frac{81000}{\frac{\Delta \theta}{10} \cdot \frac{\Delta \theta}{10} \cdot \Delta \theta} < \frac{81000}{\Delta x \Delta y \Delta \theta} < 2000000 \quad (3-14)$$

求解式(3-14)，可得到

$$\Delta \theta > 1.6^\circ \quad (3-15)$$

经过反复试验发现，效果较好的精度选择结果为

$$\begin{cases} \Delta x = 0.1m \\ \Delta y = 0.1m \\ \Delta \theta = 5^\circ \end{cases} \quad (3-16)$$

此时状态空间大小为 162000，相对而言处于可接受范围内。

### 3.3.3 状态空间简化

经过 3.3.2 节的对状态空间的设计之后，状态空间的大小依旧在百万数量级，较为庞大。然而对问题场景进行进一步分析，发现存在如下几处可能可以减少状

态空间大小的潜在机会，以从  $y$  轴负半轴向  $x$  轴负半轴行驶的左转弯车辆为例进行说明：

(1) 在对信号灯各相位进行设计时，相向道路的左转弯相位是同时出现的，故需要在交叉口区域预留出一半的空间给相向行驶的左转弯车辆。在本例中为主对角线以上部分的区域；

(2) 在设定道路通行规则时，我们规定左转车辆只能在最靠内车道行驶，故外侧车道延伸到交叉口部分的区域也可不纳入可行状态空间；

(3) 在本例中，考虑到车辆如果行驶到左下角区域，既绕了远路又不会改善舒适度，我们有充足的理由认为车辆的最优轨迹不会经过左下角附近，故该部分区域可不被纳入状态空间；

(4) 车辆在左转弯时，在理想的轨迹上， $\theta$  的变化范围应该是  $[90^\circ, 180^\circ]$ ，故  $\theta$  不在此范围的状态空间可被忽略。

对于上述可被削减的状态空间，只需在初始化  $Q$  函数时，令该部分状态空间的初始值为  $-\infty$  即可。如此一来，经训练过后，凡是靠近该部分区域的状态空间对应的  $Q$  值均会被赋为  $-\infty$ ，在之后的训练中不会重新被选中。

将简化过后的可行状态空间的区域进行绘制，如图 3.3 中非阴影部分所示。

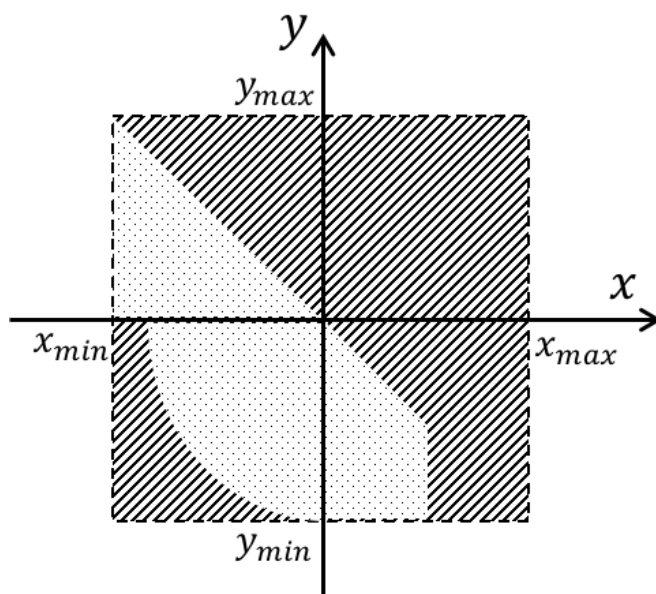


图 3.3 可行状态空间区域



在图 3.3 中，在第三象限部分，我们希望车辆离左下角距离不要太近，但第三象限仍应预留较为充足的可行区域。故将该部分可行区域的边界设置为椭圆的四分之一：

$$\begin{cases} \frac{x^2}{(|x_{min}| - \delta)^2} + \frac{y^2}{|y_{min}|^2} = 1 & (x \leq 0, y \leq 0) \\ 0 < \delta < \frac{|x_{min}|}{2} \end{cases} \quad (3-17)$$

其中  $\delta$  起到的作用是尽可能缩小可行区域的面积以达到减少可行状态空间大小的目的，但同时  $\delta$  也不能太大，否则会妨碍车辆正常的左拐轨迹。

在双向四车道的路口，如图 3.3 所示，可行状态空间对应的区域可表示为

$$\begin{cases} y \leq \frac{y_{max}}{|x_{min}|} x, & x \in (-\infty, 0], y \in (0, +\infty) \\ \frac{x^2}{(|x_{min}| - \delta)^2} + \frac{y^2}{|y_{min}|^2} \leq 1, & x \in (-\infty, 0], y \in (-\infty, 0] \\ y \leq \frac{|y_{min}|}{x_{max}} x, & x \in (0, \frac{x_{max}}{2}] \end{cases} \quad (3-18)$$

图 3.3 中可行状态空间对应的区域面积为

$$S = \frac{1}{2} |x_{min}| \cdot y_{max} + \frac{\pi}{4} (|x_{min}| - \delta) \cdot |y_{min}| + \frac{3}{8} x_{max} \cdot |y_{min}| \quad (3-19)$$

在本例中，由于路口具有中心对称性，故有性质  $|x_{min}| = x_{max} = |y_{min}| = y_{max} = l$ ，可将式(3-19)改写为

$$S = \frac{7 + 2\pi}{8} l^2 - \frac{\pi}{4} \delta l \quad (3-20)$$

另一方面，在状态空间简化前，可行状态空间对应的区域面积为

$$S_0 = (|x_{min}| + x_{max}) \cdot (|y_{min}| + y_{max}) = 4l^2 \quad (3-21)$$

在  $x, y$  方面简化后, 状态空间的大小仅为原来的

$$\frac{S}{S_0} = \frac{\frac{7+2\pi}{8}l^2 - \frac{\pi}{4}\delta l}{4l^2} = \frac{7+2\pi}{32} - \frac{\pi\delta}{16l} \quad (3-22)$$

再考虑到在  $\theta$  方面, 未简化前的范围是 $[0, 360^\circ]$ , 简化后的范围是 $[90^\circ, 180^\circ]$ , 故经过状态空间的简化后, 状态空间的大小应是原来的

$$\frac{S}{S_0} \cdot \frac{180-90}{360-0} = \frac{1}{4} \cdot \frac{S}{S_0} = \frac{7+2\pi}{128} - \frac{\pi\delta}{64l} \quad (3-23)$$

以双向四车道的交叉口为例, 根据国家标准, 标准公路的宽度为 3.75 m, 故  $|x_{min}| = x_{max} = |y_{min}| = y_{max} = 15$  m, 此外设定  $\delta = 1$  m, 可计算得到简化后, 状态的大小约为

$$1620000 \times \left( \frac{7+2\pi}{128} - \frac{\pi}{64 \times 15} \right) \approx 162814 \quad (3-24)$$

约为原状态空间大小的十分之一左右, 显著提升了运算效率。

### 3.4 策略集

在本项目中, 如 3.3.1 节所述, 为了减小状态空间, 我们未将速度、加速度信息纳入状态空间, 而是将其作为策略, 在当前状态与下一时刻状态之间的转移过程中予以体现。接下来, 我们将分析如何通过策略集确定下一时刻的状态空间的范围。

将  $t$  时刻车辆所处的位置记为  $P_t$ , 其坐标为  $(x_t, y_t)$ ,  $t$  时刻车辆的速度记为  $\mathbf{v}_t$ ,  $t$  时刻车辆的加速度记为  $\mathbf{a}_t$ , 状态之间的时间差为  $\Delta t$ 。

此外, 为简化计算, 我们作出如下假设:

- (1) 考虑到  $\Delta t$  较小, 在  $\Delta t$  时间段内, 车辆的速度近似恒定;
- (2) 车辆的最大加速度大小  $|\mathbf{a}_{max}|$  恒定;
- (3) 车辆能在  $\Delta t$  时间段内以任意方向且大小在  $[0, |\mathbf{a}_{max}|]$  之间的加速度进行速度控制。

则在求解  $P_{t+\Delta t}$  时，可列写关系式

$$\begin{cases} \overrightarrow{P_{t-\Delta t}P_t} = \mathbf{v}_{t-\Delta t}\Delta t \\ \overrightarrow{P_tP_{t+\Delta t}} = \mathbf{v}_t\Delta t \\ \mathbf{v}_t = \mathbf{v}_{t-\Delta t} + \mathbf{a}_{t-\Delta t}\Delta t \end{cases} \quad (3-25)$$

我们将式(3-25)进一步改写为

$$\overrightarrow{P_tP_{t+\Delta t}} = \mathbf{v}_t\Delta t = (\mathbf{v}_{t-\Delta t} + \mathbf{a}_{t-\Delta t}\Delta t) \cdot \Delta t = \overrightarrow{P_{t-\Delta t}P_t} + \mathbf{a}_{t-\Delta t}(\Delta t)^2 \quad (3-26)$$

将该向量关系式在平面区域进行绘制，如图 3.4 所示。

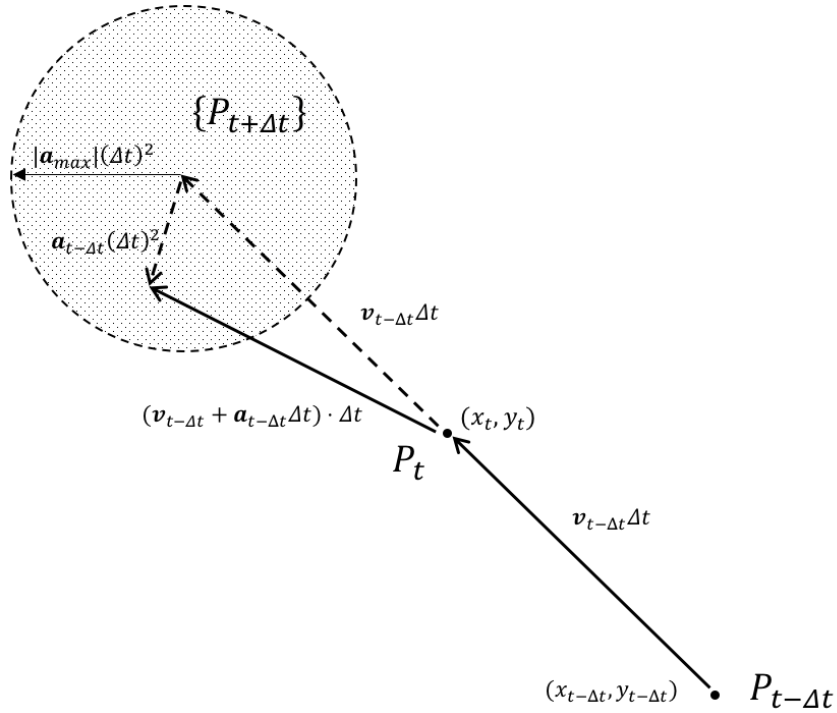


图 3.4 可能状态集合

如图 3.4 所示，根据假设，车辆能在  $[t - \Delta t, t]$  以任意方向且大小在  $[0, |\mathbf{a}_{max}|]$  之间的加速度进行速度控制，故在  $t + \Delta t$  时刻的状态空间的位置的集合  $\{P_{t+\Delta t}\}$  可由以  $|\mathbf{a}_{max}|(\Delta t)^2$  为半径的圆弧内部区域表示。

在进行数值计算时，根据(3-26)，将  $\{P_{t+\Delta t}\}$  表示为

$$\begin{cases} \{P_{t+\Delta t}\} = \{(x, y) | (x - x')^2 + (y - y')^2 \leq |a_{max}|^2 (\Delta t)^4\} \\ x' = 2x_t - x_{t-\Delta t} \\ y' = 2y_t - y_{t-\Delta t} \end{cases} \quad (3-27)$$

$\{P_{t+\Delta t}\}$  即为待求的  $t + \Delta t$  时刻的状态空间的集合。

### 3.5 奖赏函数

在 Q-学习算法中，奖赏函数直接决定了 Q 值的相对大小以及 Q 函数对不同状态的敏感性，直接决定了训练效果。

由于我们对最优轨迹的要求是在尽可能快速准确地完成转弯过程的同时具有较高的舒适性，故我们在设计奖赏函数时综合考虑了距离因素、运动方向因素和舒适度，并对三者组成奖赏函数时的参数设定进行了反复实验与调整。

#### 3.5.1 距离因素

在考虑离终点的距离因素时，以从  $y$  轴负半轴向  $x$  轴负半轴行驶的左转弯车辆为例进行说明。

在交叉口连接的路段均为双向四车道时，我们选择内侧车道作为车辆拐弯后的出口。如图 3.2 所示，在该情境下，内侧车道的中心纵坐标为  $y_{max}/4$ ，故我们选取点  $(x_{min}, y_{max}/4)$  作为终点。

在衡量离终点的距离因素为奖赏值做出的贡献时，显然应该满足与终点的距离越小，距离因素的奖赏值越大。此外，考虑到我们希望离终点距离越近，距离因素的奖赏值对距离的变化越敏感，为此，我们采用反比例函数对距离因素产生的奖赏值贡献进行刻画。

在实际应用中，我们最终设计的距离因素函数  $D(x, y)$  为

$$\begin{cases} D(x, y) = \frac{1}{\frac{d(x, y)^2}{k_d} + \zeta} \\ d(x, y) = \sqrt{(x - x_{min})^2 + \left(y - \frac{y_{max}}{4}\right)^2} \\ k_d = (x_{max} - x_{min})^2 + (y_{max} - y_{min})^2 \\ \zeta = 0.01 \end{cases} \quad (3-28)$$

其中  $d(x,y)$  为离终点的欧氏距离，为了在接近车辆终点时提高其敏感性，此处采用  $d(x,y)^2$ ， $k_d$  为归一化参数， $\zeta$  则将  $D(x,y)$  的上限限制在  $1/\zeta$  以内。

### 3.5.2 运动方向因素

在车辆运动方向的因素时，依旧以从  $y$  轴负半轴向  $x$  轴负半轴行驶的左转弯车辆为例进行说明。

在交叉口连接的路段均为双向四车道时，如图 3.2 所示，从起点到终点的线段对应的倾角为  $135^\circ$ 。直观而言，车辆运动的方向越接近  $135^\circ$  时，运动方向因素的奖赏值应该越大。考虑到我们希望越接近  $135^\circ$  时，运动方向因素的奖赏值对运动方向的变化越敏感，为此，我们同样采用反比例函数对运动方向因素产生的奖赏值贡献进行刻画。

在实际应用中，我们最终设计的距离因素函数  $\theta(\theta)$  为

$$\begin{cases} \theta(\theta) = \frac{1}{\frac{|\theta - 135|}{k_\theta} + \zeta}, & \theta \in [90, 180] \\ k = 45 \\ \zeta = 0.01 \end{cases} \quad (3-29)$$

其中  $k_\theta$  为归一化参数， $\zeta$  则将  $\theta(\theta)$  的上限限制在  $1/\zeta$  以内。

### 3.5.3 舒适度

在衡量舒适度方面，我们借鉴了 ISO 关于人体全身振动评价方法<sup>[34]</sup>，将行驶过程中，考虑人体在三维空间中各个方向受到的加速度，通过计算加速度加权均方根得到评价人体舒适度的度量。计算方法如下：

$$a_w = \sqrt{(1.4a_x)^2 + (1.4a_y)^2 + a_z^2} \quad (3-30)$$

计算出加速度加权均方根之后，参阅表 3-2 即可计算出量化的舒适度。当舒适度越小时，舒适度因素的奖赏值应该越小。考虑到我们希望在舒适程度较低时，增大因为不舒适带来的惩罚，为此，我们采用反比例函数的相反数对舒适度因素产生的奖赏值贡献进行刻画。

表 3-2 加速度、不适程度、舒适度关系

$a_w(\text{m/s}^2)$	<0.315	0.315~0.63	0.5~1.0	0.8~1.6	1.25~2.5	>2.0
不适程度	无不舒服感	一些不舒服	较不舒服	不舒服	很不舒服	极不舒服
舒适度	1	0.8	0.6	0.4	0.2	0

在实际应用中，我们最终设计舒适度函数  $A(a_w)$  为

$$\begin{cases} A(a_w) = -\frac{1}{a_w + \zeta} \\ a_w = \sqrt{(1.4a_x)^2 + (1.4a_y)^2 + a_z^2} \\ \zeta = 0.01 \end{cases} \quad (3-31)$$

其中  $\zeta$  将  $A(a_w)$  的上限限制在  $1/\zeta$  以内。

#### 3.5.4 参数设置

综合考虑各方面因素，奖赏函数可被表示为

$$R(x, y, \theta, a_w) = \lambda_d D(x, y) + \lambda_\theta \theta(\theta) + \lambda_a A(a_w) \quad (3-32)$$

只需确定权重  $\lambda_d$ ,  $\lambda_\theta$ ,  $\lambda_a$  即可确定奖赏函数形式。由于距离因素与运动方向因素所起到的作用比较相近，考虑对称性，可令

$$\lambda_d = \lambda_\theta \quad (3-33)$$

而在确定  $\lambda_a$  时，如果  $\lambda_a$  过小，舒适度因素起到的作用较弱，可能会出现加速度过大的情况，极大地影响乘客舒适度甚至在实际汽车中难以实现；而  $\lambda_a$  过大时，经过仿真测试，发现此时 Q-学习算法在可行状态空间中的训练更加充分，但路线容易出现绕弯的现象，显然不是最优路线。可能原因是，当  $\lambda_a$  较大时，Q 函数更新值会比较小，甚至接近零值，导致智能体在选择行动策略时更容易出现“过度探索”的情况，在非最优解上进行过度训练，反而使得训练结果陷入非最优的情况。经过试验， $\lambda_d$ ,  $\lambda_\theta$ ,  $\lambda_a$  满足如下条件时，训练效果较好：

$$\frac{1}{2}\lambda_a < \lambda_d = \lambda_\theta < 2\lambda_a \quad (3-34)$$

经过反复尝试，比较理想的一组参数是

$$\lambda_d = \lambda_\theta = \lambda_a = 1 \quad (3-35)$$

此时，式(3-32)为

$$R(x, y, \theta, a_w) = D(x, y) + \theta(\theta) + A(a_w) \quad (3-36)$$

其中  $D(x, y)$ ， $\theta(\theta)$ ， $A(a_w)$  的定义分别参见式(3-28)，(3-29)，(3-31)。

### 3.6 优化结果

在按照 3.2、3.3、3.4 和 3.5 节设定 Q-学习算法之后，我们进行了大量的仿真实验。我们将对优化结果进行详细的说明与分析，并将得到的轨迹进行抽象化与数学模型化，证明了其有效性并分析了其可行性。

#### 3.6.1 仿真轨迹

我们获得的一条优化效果较好的轨迹如图 3.5 中蓝线所示。

图 3.5 中红色区域是在利用 3.3.3 节中的方法对状态空间进行简化时，被设定为不应访问的空间。

分析该轨迹，可见，该轨迹具有如下性质：

- (1) 轨迹的主要部分为直线，从起点指向终点，路程长度较短，有助于缩短拐弯过程的耗时；
- (2) 在轨迹的起始部分，车辆会先平缓地拐弧形弯，以尽可能减小车辆加速度，照顾乘客的舒适度。

注意到，在轨迹终点附近，车辆运动方向从  $135^\circ$  拐向  $180^\circ$  时，运动过程并未做到平滑过渡，而是直接进行急拐弯，使得该部分的轨迹不尽理想。可能原因在于，一方面，在设定状态空间时，车辆的物理信息被离散化，在优化的时候会带来一定的偏差；另一方面，由于状态空间较为庞大，即使在现有计算条件下

进行了较为充分的训练，也只能覆盖到 25% 左右的空间，离全局最优解还有一定距离。

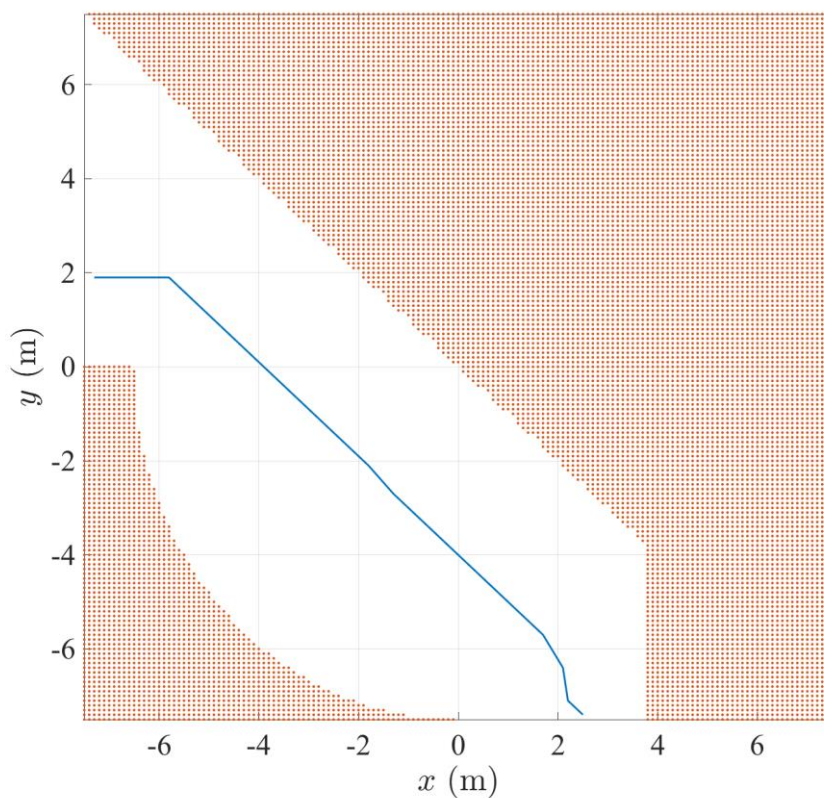


图 3.5 轨迹优化结果图

### 3.6.2 最优轨迹设计

目前的训练结果已经揭示了最优轨迹的内在规律：

- (1) 在起始时以圆弧为轨迹，使车辆运动方向从  $90^\circ$  拐向  $135^\circ$ ；
- (2) 在起始时拐弯后，沿着倾角为  $135^\circ$  的直线向终点行驶；
- (3) 在结束时以圆弧为轨迹，使车辆运动方向从  $135^\circ$  拐向  $180^\circ$ 。

根据以上规律，以从  $y$  轴负半轴向  $x$  轴负半轴行驶的左转弯车辆为对象，可设计左转轨迹如图 3.6 所示。



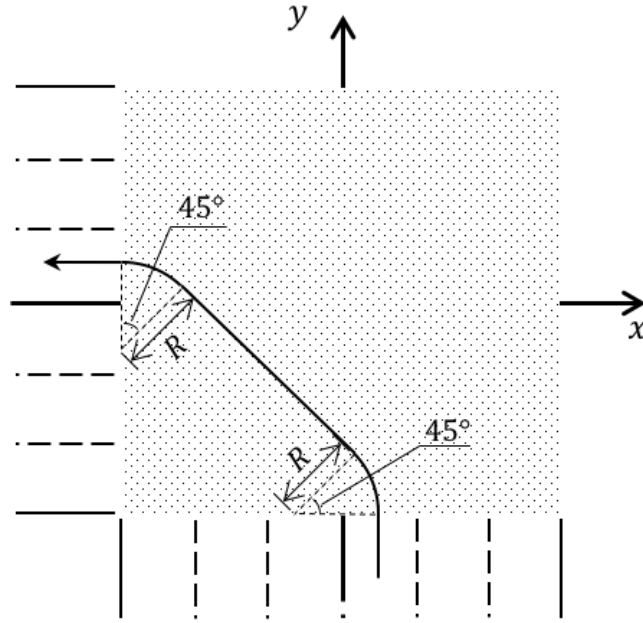


图 3.6 轨迹设计示意图

假设单条道路宽度为  $D$ ，转弯半径为  $R$ ，路口横纵方向均为双向  $2n$  车道，车辆运动方向从  $90^\circ$  拐向  $135^\circ$  的轨迹和使车辆运动方向从  $135^\circ$  拐向  $180^\circ$  的轨迹均为八分之一圆弧，则完整的左转弯轨迹可分段地写为

$$\left\{ \begin{array}{l} x = \frac{D}{2}, \quad y \in (-\infty, -nD] \\ \left(x - \frac{D}{2} + R\right)^2 + (y + nD)^2 = R^2, \quad x \in \left[\frac{D}{2} - \frac{2 - \sqrt{2}}{2}R, \frac{D}{2}\right) \\ y = -x + (\sqrt{2} - 1)R - \frac{2n - 1}{2}D, x \in \left[-nD + \frac{\sqrt{2}}{2}R, \frac{D}{2} - \frac{2 - \sqrt{2}}{2}R\right) \\ (x + nD)^2 + \left(y - \frac{D}{2} + R\right)^2 = R^2, \quad x \in \left[-nD, -nD + \frac{\sqrt{2}}{2}R\right) \\ y = \frac{D}{2}, \quad x \in (-\infty, -nD) \end{array} \right. \quad (3-37)$$

### 3.6.3 轨迹有效性验证

考虑到定义域的有效性， $n$ 、 $D$  和  $R$  之间应该满足

$$\frac{D}{2} - \frac{2 - \sqrt{2}}{2} R > -nD + \frac{\sqrt{2}}{2} R \quad (3-38)$$

将式(3-38)进行改写，可得到

$$R < \left(n + \frac{1}{2}\right) D \quad (3-39)$$

根据国家标准，标准公路的宽度为 3.75 m，在考虑到我们研究的交叉口规模最小的是双向四车道，有  $n \geq 2$ ，故

$$\left(n + \frac{1}{2}\right) D \geq \left(2 + \frac{1}{2}\right) \times 3.75 = 9.375 \text{ m} \quad (3-40)$$

而我们所研究的无人车属于小型车，其最小转弯半径约为 6 m 左右，故优化的轨迹是可满足车辆通行需求的。

式(3-39)满足时，轨迹长度为

$$\begin{aligned} L &= \frac{1}{4} \cdot 2\pi R + \sqrt{2} \left[ \left( \frac{D}{2} - \frac{2 - \sqrt{2}}{2} R \right) - \left( -nD + \frac{\sqrt{2}}{2} R \right) \right] \\ &= \frac{\pi - 2\sqrt{2}}{2} R + \sqrt{2} \left( n + \frac{1}{2} \right) D \end{aligned} \quad (3-41)$$

而传统的转弯轨迹可被近似为四分之一圆弧，其长度为

$$L_0 = \frac{1}{4} \cdot 2\pi \cdot \left(n + \frac{1}{2}\right) D = \frac{\pi}{2} \left(n + \frac{1}{2}\right) D \quad (3-42)$$

综合式(3-39)、(3-41)和(3-42)，有

$$\begin{aligned} \frac{L}{L_0} &= \frac{\frac{\pi - 2\sqrt{2}}{2} R + \sqrt{2} \left(n + \frac{1}{2}\right) D}{\frac{\pi}{2} \left(n + \frac{1}{2}\right) D} \\ &= \frac{2\pi - 4\sqrt{2}}{(2n + 1)\pi} \cdot \frac{R}{D} + \frac{2\sqrt{2}}{\pi} \end{aligned} \quad (3-43)$$

$$< \frac{2\pi - 4\sqrt{2}}{(2n+1)\pi} \cdot \frac{\left(n + \frac{1}{2}\right)D}{D} + \frac{2\sqrt{2}}{\pi} = 1$$

综合式(3-40)和(3-43)，优化后的轨迹长度小于传统转弯轨迹，且优化轨迹能满足车辆的运动性能要求。

## 3.7 性能分析

以下将从覆盖率和训练耗时两方面对 Q-算法的性能进行分析。

### 3.7.1 覆盖率

我们将“覆盖率”定义为：在训练过程中，智能体访问过的状态空间数目在总状态空间数目中的比例。

在本项目中，由于问题的固有性质，状态空间的大小比较大。即使通过 3.3.3 节中的方法对状态空间进行简化，状态空间的大小依旧在十万数量级，难以通过训练将其完全覆盖。故在分析 Q-学习算法的算法性能时，在有限的训练次数内能探索并覆盖的状态空间的数量将成为十分关键的指标。

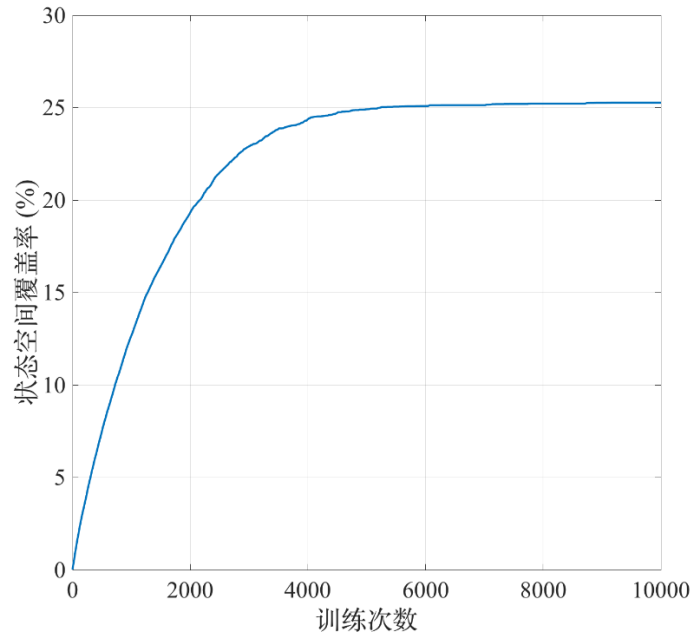


图 3.7 状态空间覆盖率与训练次数关系曲线图

我们在 Q-学习算法进行训练时采集相关信息并将状态空间覆盖率与训练次数之间的关系绘制如图 3.7 所示。

可见，在训练开始时，覆盖率随训练次数的增加而快速增长，但在 5000 次之后增速显著放缓，而在 9000 次至 10000 次之间覆盖率几乎是水平线段。覆盖率最终收敛到 25% 左右。可能原因在于，一方面，根据式(3-9)可知，随着训练次数的增加， $\epsilon$  单调减小，使得智能体探索此前未访问过的状态空间的可能性降低；另一方面，随着训练逐渐趋于充分，最优路线基本得到确定，训练的随机性进一步减小。

由此可以看出，训练次数  $N \in [6000, 10000]$  时较为合适，在计算次数与覆盖率之间能取得较好的平衡。

### 3.7.2 训练耗时

衡量运算性能的另一项指标是训练耗时情况。本项目的训练过程在 MacBook Pro 2016 上进行，训练耗时测算的环境为：

- (1) 中央处理器：2.7 GHz Intel Core i7；
- (2) 内存：16 GB 2133 MHz LPDDR3；
- (3) 训练环境：MATLAB for Mac R2016b。

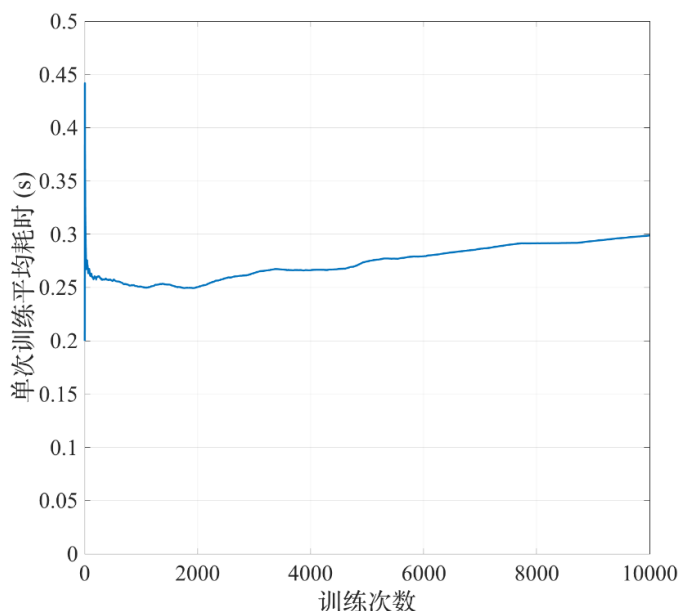


图 3.8 单次训练平均耗时与训练次数关系曲线图

我们在训练时记录相关数据，并将将单次训练平均耗时与训练次数之间的关系曲线进行绘制，如图 3.8 所示。

可见，单次训练的平均耗时随着训练次数的增加呈现先急速减少后缓慢增加的趋势，且增长趋势偏向线性。

可能原因为，在起始时，由于状态空间基本未得到训练，从智能体在路网中的行动随机性较强，单次训练可能绕路现象较为严重，耗时较多；随着训练次数的增加，较优轨迹逐渐被发现，绕路现象得到显著减轻，耗时急剧减少；在训练中后期，由于舒适度的要求，车辆的加速减速过程比较平缓，在一定程度上增加了通行时间，使得单次训练的耗时也相应增加，但由于加速减速过程在转弯过程中出现的次数较为有限，故单次训练耗时的增长也较为有限。

### 3.8 本章小结

在本章，我们利用 Q-学习算法对无人车左转弯轨迹进行优化。在理论层面，本章从原理、参数设定、状态空间设计、策略集设计和奖赏函数设计方面对 Q-学习算法在该场景下的应用进行了详细的分析与论证。此外，在本章，我们进行了仿真实验，并通过对优化结果的说明和对优化过程的性能分析进一步分析了该方法的有效性与可行性。

在理论层面，我们先是通过步骤说明与流程图对 Q-学习算法的原理进行了说明，接着结合问题本身的性质，在参数设定方面对学习速率因子、奖励折现因子、训练次数和贪婪参数进行了设定与调整。接着，我们对状态空间的设计进行了详细地说明，包括构成状态空间的参数、离散化过程的精度和根据交叉口性质对状态空间进行简化的方法。最后，我们详细描述了奖励函数的设计理念、设计方法与设计结果。

在仿真结果方面，我们通过仿真，得到了兼顾通行速度与通行舒适性的优化轨迹：在起始时平缓拐弯，朝终点直线行驶，在终点附近再平滑拐弯。我们接着将该优化结果抽象为数学形式，并论证了其有效性与可行性。我们最后利用仿真过程中记录的数据，说明了本算法在性能方面具有较好的实际可行性。

## 第4章 多车协同策略优化

此前，我们利用 Q-学习算法，以单个无人车为研究对象，优化了无人驾驶车辆在交叉口左转弯时的轨迹。在本章节，我们将 Q-学习算法应用在多辆车之间的协同策略优化。

在本章节，我们将先介绍 Q-学习算法的在本场景下进行应用的思路，接着对参数设定、状态空间设计、策略集设计、奖赏函数设计等环节进行详细分析，最后给出包括训练性能、优化效果等指标的仿真结果。

### 4.1 原理概述

在 3.1 节，我们已经详细介绍了 Q-学习算法的原理，在此处将其省略。在本节，我们将着重介绍如何将 Q-学习算法应用于多车之间的协同问题。

我们曾设想过采用多智能体的增强学习算法进行研究，但现有的较为成熟的多智能体增强学习算法所研究的多目标对象是时不变的。而在本问题中，行驶在特定的车道上的车辆是随时间变化的，难以直接应用传统的多智能体增强学习的方法。

为此我们考虑将车辆进行分组，以一组车辆的整体为研究对象，使得该组内车辆各自的状态与它们之间的位置关系共同构成状态空间，如此即可将问题转化为单智能体优化问题，进而可以用 Q-学习算法进行求解。

具体而言，在设计分组机制时，我们要考虑如下因素：

- (1) 一组内车辆数量越少越容易进行分组编队，可控性也越强；
- (2) 由于普通车辆依旧由司机驾驶，车辆协同方法能控制的车辆只有无人车；
- (3) 普通车不会考虑后方车辆。

根据以上因素，我们将分组的规模大小设定为 2 辆车。此时，共有“无人车+无人车”、“无人车+普通车”、“普通车+普通车”、“普通车+无人车”四种可能的编队方法。考虑到普通车不能作为直接控制对象，“普通车+普通车”组合不是本项目的研究对象。再考虑到普通车不会考虑后方车辆，“普通车+无人车”组合也不应考虑，因为此时无人车并不能影响到普通车。因此，我们最终可能有两种分组类型：“无人车+无人车”和“无人车+普通车”。

## 4.2 参数设定

如3.2节所述,Q-学习算法中的普适的参数设定主要包括学习速率因子  $\alpha_t$  , 奖励折现因子  $\gamma$  , 训练次数  $N$  和  $\epsilon$ -贪婪方法中的  $\epsilon$  参数。而在本问题中, 由于训练结果对状态转移时间  $\Delta t$  极为敏感, 在本节我们也会讨论转台转移时间长度的设置。

### 4.2.1 学习速率因子与奖励折现因子

如 3.2.1 节所述, 我们可构建约束条件如下:

$$\begin{cases} 1 + \alpha\gamma - \alpha > k\alpha \\ k > 1 \end{cases} \quad (4-1)$$

在 3.2.1 节中, 参数选择结果为

$$\begin{cases} \alpha = 0.5 \\ \gamma = 0.5 \\ k = 1.5 \end{cases} \quad (4-2)$$

而在本问题中, 我们在训练过程中发现, 如果按照式(4-2)进行参数设定, 训练效果不够理想: 车辆之间的速度关系曲线会出现显著波动, 甚至经常会出现由于速度控制失误, 而导致两车距离过紧, 仿真过程强行终止的情况。

可能原因在于, 对于该问题而言, 学习速率因子偏大, 导致  $Q$  值的权重较小,  $Q$  函数本身的信息未得到充分利用, 而奖赏函数的信息被过度强调, 使得训练过程更容易陷入局部最优, 导致整体优化效果不理想。

经过反复试验, 我们发现, 参数设置为

$$\begin{cases} \alpha = 0.2 \\ \gamma = 0.5 \\ k = 3 \end{cases} \quad (4-3)$$

训练效果较好。

### 4.2.2 训练次数

在本问题中, 状态空间的规模在十万左右, 理论上, 只要训练次数足够多, 所有状态都能得到遍历, 使得  $Q$  函数也将收敛到全局最优解。但在本问题中, 尽

管状态空间大小相比轨迹优化问题已经小了一个数量级，但时间成本依旧难以负担，故需要设定较为合理的训练次数，使得训练过程能在效果与效率之间达到较好的平衡。

在本问题中，设定训练次数  $N$  时，我们主要参考了训练次数与状态空间覆盖率之间的关系，最终发现  $N = 20000$  较为合适。

#### 4.2.3 贪婪参数

在 3.2.3 节中，我们已经详细说明了设定贪婪参数时所需遵循的规则，并最终采取了如下的形式对贪婪参数  $\epsilon$  进行设定：

$$\epsilon_{n+1} = \delta \cdot \epsilon_n \quad (0 < \delta < 1) \quad (4-4)$$

在 3.2.3 节中，在  $N = 10000$  时，参数选取结果为

$$\begin{cases} \epsilon_0 = 0.2 \\ \delta = 0.999 \end{cases} \quad (4-5)$$

而在优化多车协同策略时，训练次数的设定为  $N = 20000$ ，故需要对衰减因子进行再次选取。为了保证最终的衰减效果一致，我们希望在训练结束时，衰减因子的大小相等，亦即

$$\epsilon_0 \cdot 0.999^{10000} = \epsilon_0 \cdot \delta^{20000} \quad (4-6)$$

解之，可得到衰减因子的参考值为

$$\delta = \sqrt[2]{0.999} \approx 0.9995 \quad (4-7)$$

但在实际训练中，我们发现，对于多车协同策略优化问题而言，由于问题性质限制，状态更新函数对空间的搜索能力较为有限，需要在选择行动时增强随机性。为此，我们需要选择比式(4-7)更大的衰减因子，以减缓衰减速度。经过反复试验，最终发现以下参数设定

$$\begin{cases} \epsilon_0 = 0.2 \\ \delta = 0.9999 \end{cases} \quad (4-8)$$



能取得较好的效果。

#### 4.2.4 状态转移时间

由于仿真部分在 Q-学习算法中具有相当重要的作用，而状态与状态之间的转移时间  $\Delta t$  是仿真中十分重要的一项参数，故  $\Delta t$  对于训练效果也具有较为重大的影响。状态转移时间的设定规则较为复杂，既不能过大，也不能过小。进行分析如下。

当状态转移时间较大时，在一个时间周期内，车辆在加速/减速的范围会更宽，使得在状态转移过程中，对下一时刻的可能状态的搜索范围变大，进而增加了训练时间成本，不利于进行充分训练。例如，假设加速度的最大值为  $a$ ，前车速度为  $v_1$ ，后车速度为  $v_2$ ，两车间距为  $d$ ，那么在一个状态转移周期内，各个状态的最大变化量分别为

$$\begin{cases} \Delta v_1 = a\Delta t - (-a)\Delta t = 2a\Delta t \\ \Delta v_2 = a\Delta t - (-a)\Delta t = 2a\Delta t \\ \Delta d = \max\{v_1 - v_2\}\Delta t + \frac{1}{2}a(\Delta t)^2 - \left(-\frac{1}{2}a(\Delta t)^2\right) \\ \quad = \max\{v_1 - v_2\}\Delta t + a(\Delta t)^2 \end{cases} \quad (4-9)$$

而当状态转移时间变为  $k\Delta t$  时，在一个状态转移周期内，各个状态的最大变化量则变为

$$\begin{cases} \Delta v'_1 = 2ak\Delta t = k\Delta v_1 \\ \Delta v'_2 = 2ak\Delta t = k\Delta v_2 \\ \Delta d' = \max\{v_1 - v_2\}k\Delta t + a(k\Delta t)^2 \approx k^2\Delta d \end{cases} \quad (4-10)$$

按照 4.3 节的设计，当状态空间是由前后车辆的速度以及两车之间的间距组成时，若状态转移时间变为  $k\Delta t$ ，则在状态进行转移时，状态搜索空间的大小与原空间大小的比值为

$$\frac{\Delta v'_1 \cdot \Delta v'_2 \cdot \Delta d'}{\Delta v_1 \cdot \Delta v_2 \cdot \Delta d} \approx k^4 \quad (4-11)$$

由此可见，当状态转移时间变长时，进行状态搜索的时间代价将会以显著增加，例如，状态转移时间变为原来的 5 倍时，进行状态搜索所需的时间将会变为原来的 600 倍左右，将极大地影响训练的速度与效率。

此外，由于速度控制方案是基于给定的状态转移时间的，当状态转移时间较长时，训练得到的速度控制方案更为粗糙，进而降低多车协同策略的实用性。

而另一方面，由于车辆的加加速度是受车辆本身物理特性限制的，在状态转移时间较短时，车辆的加加速度会较大，可能诱导策略在物理上是不可实现的。因此，状态转移时间也不宜过短。

经过反复试验，状态转移时间设定为

$$\Delta t = 0.2 \text{ s} \quad (4-12)$$

较为合适。

## 4.3 状态空间

在增强学习中，状态空间的设计起到了至关重要的作用。在设计状态空间时，需要根据问题本身的性质进行状态空间的选取。与此同时，在我们采取的 Q-学习算法中，需要对状态空间进行离散化处理，而在离散化处理时，如何在离散化得到的结果的精度与状态空间的大小之间进行权衡也是值得讨论的问题。在本节，我们将对状态空间设计时的参数选取、状态空间内参数的范围的选择以及对状态空间进行离散化处理时的精度选择进行讨论。

### 4.3.1 参数选取

在选择构成状态空间的参数时，由于在我们的 Q-学习算法中，单个智能体其实是由两辆车构成的，故在设计状态空间时我们应将两辆车各自的信息以及它们之间相互作用的信息考虑进来。参数选取结果如表 4-1 所示。

分析表 4-1，可发现，我们只选取了最为基本的三项参数作为状态空间的组成部分。

一方面，虽然车辆的位置信息也是车辆重要的运动学参数之一，但考虑到在进行多车协同时，我们只关心车辆之间的相对位置关系（以此判断是否有交通事

故的危险), 故车辆的绝对位置信息并不必要。相应地, 我们以两辆车之间的相对位置信息, 即两车之间的距离作为车辆的位置信息构成状态空间。

表 4-1 状态空间参数选择

名称	符号	单位
前车速度	$v_1$	m/s
后车速度	$v_2$	m/s
车辆间距	$d$	m

另一方面, 我们未将车辆的加速度作为状态空间的参数, 主要的考虑原因在于, 在车辆的当前状态之后, 再给定加速度, 我们就能计算出车辆下一时刻的速度与两车之间的相对位置。因此, 加速度信息可被视为策略集中的一个动作, 我们将在策略集的设计中对此进行讨论。

#### 4.3.2 范围选择

在该问题中, 我们还需指定状态空间中各参数的取值范围, 以进行离散化处理。

##### 1) 车辆速度

由于前车与后车具有一定对称性, 故前车速度与后车速度的范围应该相同。

根据《中华人民共和国道路交通安全法实施条例》第四十六条, 机动车行驶中转弯时, 最高行驶速度不得超过 30 km/h。考虑到

$$30 \text{ km/h} = 8.33 \text{ m/s} \approx 8 \text{ m/s} \quad (4-13)$$

故速度上限可设定为 8m/s。

另一方面, 由于在两车协同的场景下, 两辆车均朝正方向运动, 速度不应为负, 故速度下限可设为 0。

##### 2) 车辆距离

在双向六车道的十字路口, 车辆直行的距离在 20 m 左右。考虑到车辆之间的协同在车辆间距较小时效果较为明显, 故我们在开始打算将车辆距离的上限设定为 10 m (超出此范围则认为车辆之间不存在相互协同关系)。但在实际应用中, 我们发现, 如果车辆之间的距离上限仅为 10 m, 那么很容易出现车辆间距超出此

范围的情况，优化效果并不理想。为此，我们将车辆之间的距离上限设定为 20 m 发现优化效果有较为明显的提升。

显然，两车之间的距离下限为 0。

#### 4.3.3 精度选择

在精度选择方面，如 3.3.2 节所述，需要将各参数进行离散化处理。

假设前车速度、后车速度和两车间距的精度分别为  $\Delta v_1$ ， $\Delta v_2$ ， $\Delta d$ ，速度范围大小与两车间距范围大小分别为  $V$  和  $D$ ，则状态空间的大小为

$$|S| = \frac{V}{\Delta v_1} \cdot \frac{V}{\Delta v_2} \cdot \frac{D}{\Delta d} \quad (4-14)$$

在该问题的情境中，状态空间大小为

$$|S| = \frac{V}{\Delta v_1} \cdot \frac{V}{\Delta v_2} \cdot \frac{D}{\Delta d} = \frac{8}{\Delta v_1} \cdot \frac{8}{\Delta v_2} \cdot \frac{20}{\Delta d} = \frac{1280}{\Delta v_1 \Delta v_2 \Delta d} \quad (4-15)$$

在实际测试中，考虑到在  $N = 20000$  时的优化效果与运算速度，状态空间大小应满足

$$|S| = \frac{1280}{\Delta v_1 \Delta v_2 \Delta d} < 200000 \quad (4-16)$$

考虑到将两车分至同一组进行优化时，两车运动状态应具有对称性，故引入约束

$$\Delta v_1 = \Delta v_2 = \Delta v \quad (4-17)$$

此时，关系式可改写为

$$|S| = \frac{1280}{(\Delta v)^2 \cdot \Delta d} < 200000 \quad (4-18)$$

经过反复试验发现，效果较好的精度选择结果为

$$\begin{cases} \Delta v_1 = 0.1 \text{ m/s} \\ \Delta v_2 = 0.1 \text{ m/s} \\ \Delta d = 0.1 \text{ m} \end{cases} \quad (4-19)$$

此时状态空间大小为 128000，处于可接受范围内。

## 4.4 策略集

在本项目中，我们采用的思路时，给定当前状态之后，根据加速度与速度的限制，计算下一时刻可能的状态空间。因此，在该计算过程中，我们将智能体的动作选择隐式地在状态转移过程中予以体现。

在本节，我们将讨论如何根据当前状态和动力学参数限制计算下一时刻可能的状态。由于“无人车+无人车”与“无人车+普通车”两种情况下车辆协同的机理有所差异，故我们将对两种情况分别进行讨论。

### 4.4.1 无人车+无人车

在“无人车+无人车”的情况下，两辆车的速度都是我们所能控制的。而另一方面，考虑到状态转移时间较短，我们可将车辆的加速过程近似为匀加速运动

$$\begin{cases} v'_1 = v_1 + a\Delta t \\ v'_2 = v_2 + a\Delta t \\ a_{min} \leq a \leq a_{max} \end{cases} \quad (4-20)$$

为了找出下一时刻所有可能的状态空间，我们采取如下的步骤：

(1) 以前车为研究对象，在离散化的空间中，遍历所有可能的加速度  $a$ ，再根据  $v'_1 = v_1 + a\Delta t$  计算下一时刻前车的速度；

(2) 以后车为研究对象，在离散化的空间中，遍历所有可能的加速度  $a$ ，再根据  $v'_2 = v_2 + a\Delta t$  计算下一时刻后车的速度；

(3) 针对每一项  $v'_1$  和  $v'_2$  的组合，分别按照匀加速过程下路程的计算方法更新两车之间的距离

$$d' = d + \left( \frac{v_1 + v'_1}{2} \right) \cdot \Delta t - \left( \frac{v_2 + v'_2}{2} \right) \cdot \Delta t \quad (4-21)$$

如此一来，下一时刻的可能状态空间集合就可表示成  $\{v'_1, v'_2, d'\}$ 。

#### 4.4.2 无人车+普通车

在“无人车+普通车”的情况下，我们只能控制无人车，而对于普通车，我们只能采用跟驰模型进行模拟。一方面，考虑到状态转移时间较短，我们可将无人车车辆的加速过程近似为匀加速运动。另一方面，在跟驰模型方面，考虑到计算代价，我们采用线性跟驰模型。因此，前车和后车的速度更新方法可总结为

$$\begin{cases} v'_1 = v_1 + a \cdot \Delta t \\ v'_2 = v_2 + \alpha \cdot (v_1 - v_2) \cdot \Delta t \\ a_{min} \leq a \leq a_{max} \end{cases} \quad (4-22)$$

其中  $\alpha$  为灵敏度，在此处取  $\alpha = 1$ 。

为了找出下一时刻所有可能的状态空间，我们采取如下的步骤：

(1) 以前车为研究对象，在离散化的空间中，遍历所有可能的加速度  $a$ ，再根据  $v'_1 = v_1 + a\Delta t$  计算下一时刻前车的速度；

(2) 以后车为研究对象，根据  $v'_2 = v_2 + \alpha \cdot (v_1 - v_2) \cdot \Delta t$  计算下一时刻后车的速度；

(3) 针对每一项  $v'_1$  和  $v'_2$  的组合，分别按照匀加速过程下路程的计算方法更新两车之间的距离

$$d' = d + \left( \frac{v_1 + v'_1}{2} \right) \cdot \Delta t - \left( \frac{v_2 + v'_2}{2} \right) \cdot \Delta t \quad (4-23)$$

如此一来，下一时刻的可能状态空间集合就可表示成  $\{v'_1, v'_2, d'\}$ 。

### 4.5 奖赏函数

在增强学习中，奖赏函数具有极为重要的作用：一方面，奖赏函数直接决定了 Q 值的相对大小，进而直接影响最终求得的 Q 函数的性质；另一方面，奖赏函数的众多性质（陡峭性、上下界）将直接影响收敛过程与运算时的性能。为此，

在设计奖赏函数时，需要根据问题性质进行设计。在本节，我们将介绍在优化多车协同问题时考虑的因素以及尝试的函数形式。

#### 4.5.1 速度因素

在进行多车协同问题的优化时，我们首要的目标是保障安全性。在保证车辆协同行驶方案安全可靠的前提下，我们希望能使得车辆整体的行驶速度更快。考虑到车辆间距是由速度决定的，故我们先分析设计较为重要的速度因素。

容易想到，两车之间较为理想的速度设定结果是两车能以相同的速度匀速行驶，如此一来，能保证两车之间几乎不会相撞。而另一方面，考虑到我们的协同对象仅为两辆车，为了保证其它车也不至于与作为优化对象的两辆车相撞，我们希望能在设置奖赏函数的速度因素时，能引导所有车辆的速度趋近于同一个较优的速度限定值。

在设计速度因素对奖赏值做出的贡献时，一个显然的要求是，当前速度与目标速度的距离越小，速度因素的奖赏值越大。此外，考虑到我们希望离目标速度的距离越近，速度因素的奖赏值对速度的变化越敏感，为此，我们采用反比例函数对速度因素产生的奖赏值贡献进行刻画。

在实际应用中，我们以  $i = 1$  表示前车，以  $i = 2$  表示后车，则最终设计的速度因素函数  $V(v_i)$  为

$$\left\{ \begin{array}{l} V(v_i) = \frac{1}{\frac{|v_i - v^*|}{k_v} + \zeta} \\ k_v = \max\{v^* - v_{min}, v_{max} - v^*\} \\ v^* = 5 \\ \zeta = 0.1 \end{array} \right. \quad (4-24)$$

其中  $v_i$  为车辆  $i$  当前的速度， $v^*$  为我们希望的所有车辆趋向的最优速度限定值。 $k_v$  为归一化参数， $\zeta$  则将  $D(x, y)$  的上限限制在  $1/\zeta$  以内。

值得一提的是，我们之所以设置  $v^* = 5$ ，是出于如下考虑：

(1) 考虑到在实际行驶过程中，可能会出现紧急情况需要紧急制动。而速度为 5 m/s 时，车辆紧急制动的制动距离较小，在安全性上较有保障；

(2) 速度为 5 m/s 时，车辆行驶速度并不低，交叉口通行时间不至于过长，能在通行效率上维持较高的水准。

#### 4.5.2 距离因素

我们也曾考虑将两车之间的间距作为距离因素加以考虑。

在衡量两车间距为奖赏值做出的贡献时，我们希望距离因素对奖赏值的影响如下：

(1) 当两车间距过小时，考虑到此时制动距离有限，容易发生追尾事故等。故两车间距小于临界安全值时，奖赏值应减小；

(2) 当两车间距过大时，此时交叉口通行能力下降，车辆通过交叉口的效率也会降低。故两车间距大于临界值时，奖赏值也应减小；

(3) 当两车间距小于临界安全值时，两车间距越小，距离因素的奖赏值对距离的变化越敏感；当两车间距大于临界值时，两车间距越大，距离因素的奖赏值对距离的变化越敏感。

根据以上原则，我们最终设计的距离因素函数  $D(d)$  为

$$\begin{cases} D(d) = \frac{1}{l(d) + \zeta} \\ \zeta = 0.01 \end{cases} \quad (4-25)$$

其中， $\zeta$  将  $D(d)$  的上限限制在  $1/\zeta$  以内，而  $l(d)$  可用分段函数的形式表示

$$l(d) = \begin{cases} \frac{d}{\underline{d}}, d \in [0, \underline{d}) \\ \infty, d \in [\underline{d}, \bar{d}] \\ \frac{d_{max} - d}{d_{max} - \bar{d}}, d \in (\bar{d}, d_{max}] \end{cases} \quad (4-26)$$

其中  $\underline{d}$  为距离临界安全值， $\bar{d}$  为较优距离上界。

经过反复尝试，我们发现参数取

$$\begin{cases} \underline{d} = 3 \\ \bar{d} = 7 \end{cases} \quad (4-27)$$

效果较好。

分析  $D(d)$ ，可发现其规律如下：



- (1)  $d \in [0, \underline{d})$  时, 随着  $d$  的增加,  $D(d)$  由  $1/\zeta$  单调下降, 逐渐趋于  $1/(1 + \zeta)$ ;
- (2)  $d \in [\underline{d}, \bar{d}]$  时,  $D(d)$  始终为 0;
- (3)  $d \in (\bar{d}, d_{max}]$  时, 随着  $d$  的增加,  $D(d)$  由  $1/(1 + \zeta)$  单调增加, 逐渐趋于  $1/\zeta$ 。

#### 4.5.3 函数形式

综合考虑各方面因素, 奖赏函数可被表示为

$$R(v_1, v_2, d) = \lambda_{v1}V(v_1) + \lambda_{v2}V(v_2) + \lambda_d D(d) \quad (4-28)$$

只需确定权重  $\lambda_{v1}$ ,  $\lambda_{v2}$ ,  $\lambda_d$  即可确定奖赏函数形式。考虑到前车和后车的对称性, 可令

$$\lambda_{v1} = \lambda_{v2} \quad (4-29)$$

根据设计, 两车间距越不理想时,  $D(d)$  越大, 因此,  $\lambda_d$  应为负数。但在训练过程中, 我们发现, 在设计奖赏函数时, 距离因素极难控制, 甚至会影响到对速度的训练。原因在于, 有时在控制速度时, 会出现超调等现象, 导致距离短暂地超出较优距离范围。但由于惩罚作用的存在, 车辆会避免这些情况的出现, 导致速度控制结果不尽如人意。

因此, 我们尝试将距离因素从奖赏函数的结构中移除, 发现训练效果得到了提升。为此, 在我们最终设计的奖赏函数中,  $\lambda_d = 0$ 。考虑到只有  $\lambda_{v1}$  和  $\lambda_{v2}$  存在, 且式(4-29)确定了  $\lambda_{v1}$  和  $\lambda_{v2}$  的相对关系, 故我们最终将参数设定为

$$\begin{cases} \lambda_{v1} = \lambda_{v2} = 1 \\ \lambda_d = 0 \end{cases} \quad (4-30)$$

在确定相对关系后, 我们在设计  $V(v_i)$  时也曾采取其他形式的函数, 如 -2 次幂函数和指数函数, 但训练效果并不好。

在采取 -2 次幂函数时, 训练得到的多车协同方案中, 在稳定时, 车辆速度经常呈现锯齿状波动, 即前后两车速度呈现互补锯齿状。尽管在此种情况下, 两车

间距能维持稳定，但频繁的加速、减速并不现实也不必要，该方案显然不是最优方案。

而采用指数函数的形式时， $Q$  函数中的  $Q$  值容易过大，容易导致对新的状态空间的探索能力下降。

### 4.6 优化结果

我们分别针对“无人车+无人车”和“无人车+普通车”的情景进行仿真，并在本节介绍训练得到的优化结果。

在两种情景下，我们均对 4 种情况进行了仿真：

- (1) 前车初速度较小、后车初速度较小；
- (2) 前车初速度较小、后车初速度较大；
- (3) 前车初速度较大、后车初速度较小；
- (4) 前车初速度较大、后车初速度较大。

#### 4.6.1 无人车+无人车

在“无人车+无人车”的情况下，四种情况的仿真结果分别如下所述。

- 1) 前车初速度较小、后车初速度较小

该种情况可被视作两辆车的同步加速情景。

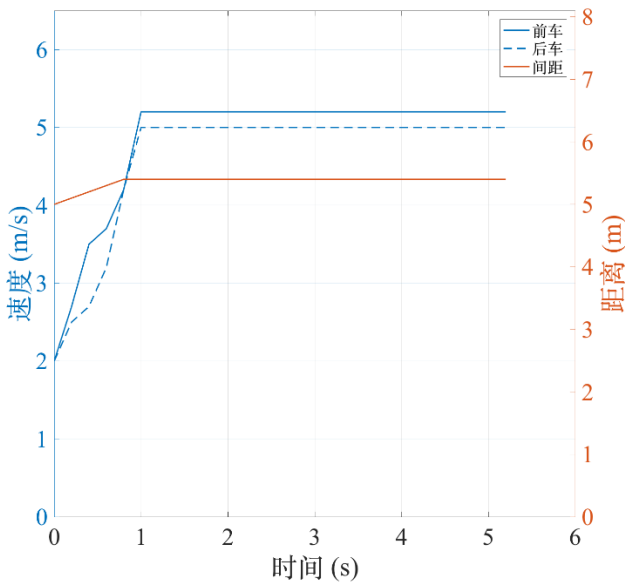


图 4.1 “无人车+无人车”情况 1 仿真结果

我们将前车和后车的初速度均指定为 2 m/s，进行仿真，结果如图 4.1 中的双轴折线图所示。

分析图 4.1，可见前车和后车的速度均从最开始的 2 m/s 较为迅速地增大到我们设定的最优速度 5 m/s，并在此后保持速度与距离的稳定。值得注意的是，最终稳定时，两车速度并不完全相等。原因在于，前后车速度差造成的距离变化小于离散化的精度范围，使得距离维持不变，进而导致该状态被认定为稳定状态。但在实际应用中，我们可以加以改进，将前后车的速度设定为完全相同。

分析该协同过程，可以发现由于计算能力有限，我们的训练还不够充分完善。理论上由于前车和后车的初速度相同，两车应以相同的方式完成加速过程。但在仿真结果中，我们能看见，前车和后车的速度增加方式有较为细微的不同：前车速度的增加相比后车更为迅速，导致两车间距有所增加。该现象对协同效果的影响极为轻微。

## 2) 前车初速度较小、后车初速度较大

该种情况可被视作后车减速、前车加速的情景。

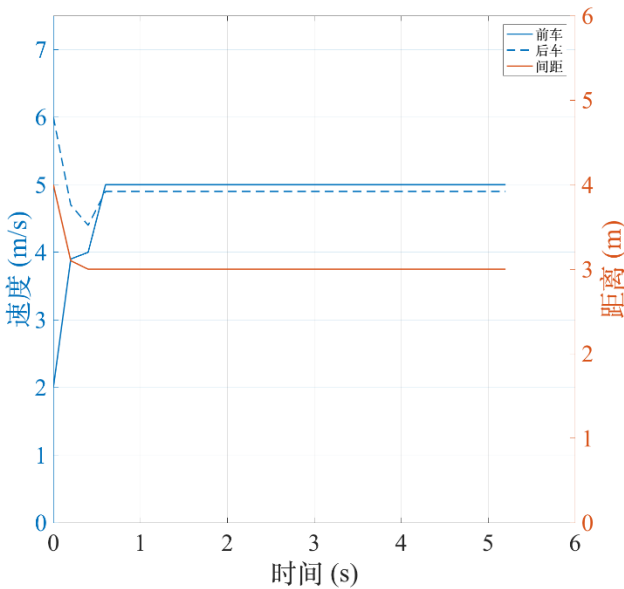


图 4.2 “无人车+无人车”情况 2 仿真结果

我们将前车初速度指定为 2 m/s，后车初速度指定为 6 m/s，进行仿真，结果如图 4.2 中的双轴折线图所示。

分析图 4.2, 可见前车速度从最开始的  $2\text{ m/s}$  较为迅速地增大到我们设定的最优速度  $5\text{ m/s}$ , 而后车速度则出现了类似超调的现象: 从最开始的  $6\text{ m/s}$  下降到最优速度以下, 再恢复到最优速度  $5\text{ m/s}$ 。在达到最优速度后, 两车均保持速度与距离的稳定。值得注意的是, 最终稳定时, 两车速度并不完全相等。原因在于, 前后车速度差造成的距离变化小于离散化的精度范围, 使得距离维持不变, 进而导致该状态被认定为稳定状态。但在实际应用中, 我们可以加以改进, 将前后车的速度设定为完全相同。

分析该协同过程, “超调”现象出现的原因可能在于, 开始时空车速度远大于前车速度, 如果后车不先将速度降低到最优速度以下, 那么很可能两车距离会过近甚至相撞。故后车通过“超调”过程保证了两者之间的安全距离。

### 3) 前车初速度较大、后车初速度较小

该种情况可被视作后车加速、前车减速的追及过程。

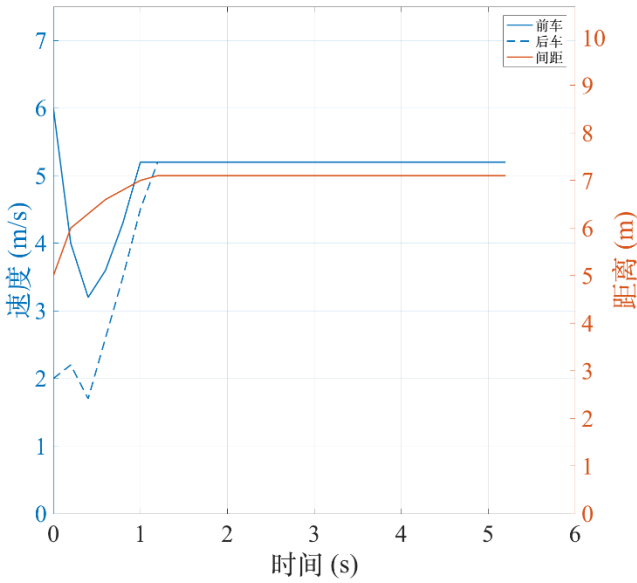


图 4.3 “无人车+无人车”情况 3 仿真结果

我们将前车初速度指定为  $6\text{ m/s}$ , 后车初速度指定为  $2\text{ m/s}$ , 进行仿真, 结果如图 4.3 图 4.1 中的双轴折线图所示。

分析图 4.3, 可见前车速度则出现了类似超调的现象: 从最开始的  $6\text{ m/s}$  下降到最优速度以下, 再恢复到最优速度  $5\text{ m/s}$ ; 后车速度则在经历了一点小波折后,

较为迅速地增大到我们设定的最优速度 5 m/s。在达到最优速度后，两车均保持速度与距离的稳定。

分析该协同过程，后车在开始时出现曲线波折的原因可能在于，计算能力有限，我们的训练还不够充分完善；而前车“超调”现象出现的原因可能在于，开始时前车速度远大于后车速度，如果前车不先将速度降低到最优速度以下，那么很可能两车距离会过远，影响道路通行效率。

4) 前车初速度较大、后车初速度较大

该种情况可被视作两辆车的同步减速情景。

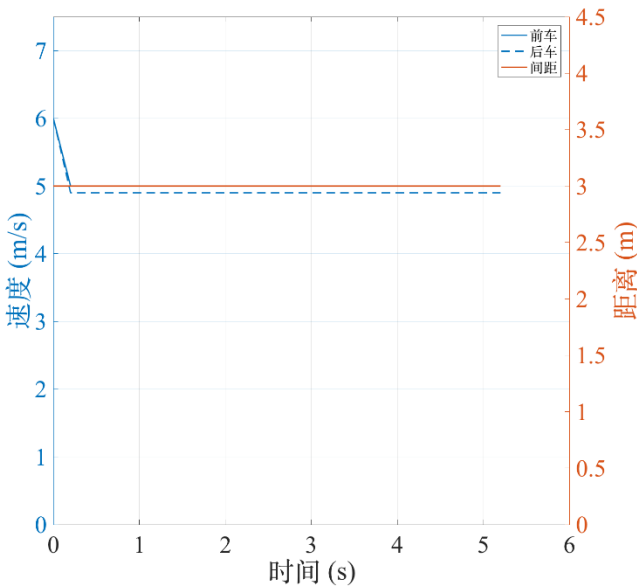


图 4.4 “无人车+无人车”情况 4 仿真结果

我们将前车和后车的初速度均指定为 6 m/s，进行仿真，结果如图 4.1 中的双轴折线图所示。

分析图 4.4，可见前车和后车的速度均从最开始的 6 m/s 较为迅速地减小到我们设定的最优速度 5 m/s，并在此后保持速度与距离的稳定。值得注意的是，最终稳定时，两车速度并不完全相等。原因在于，前后车速度差造成的距离变化小于离散化的精度范围，使得距离维持不变，进而导致该状态被认定为稳定状态。但在实际应用中，我们可以加以改进，将前后车的速度设定为完全相同。

分析该协同过程，可以发现，前车与后车的减速过程几乎一样，两车的速度曲线几乎重合，符合直觉，也说明利用增强学习的方法能获得较为理想的多车协同方法。

#### 4.6.2 无人车+普通车

在“无人车+普通人车”的情况下，四种情况的仿真结果分别如下所述。

##### 1) 前车初速度较小、后车初速度较小

该种情况可被视作两辆车的同步加速情景。

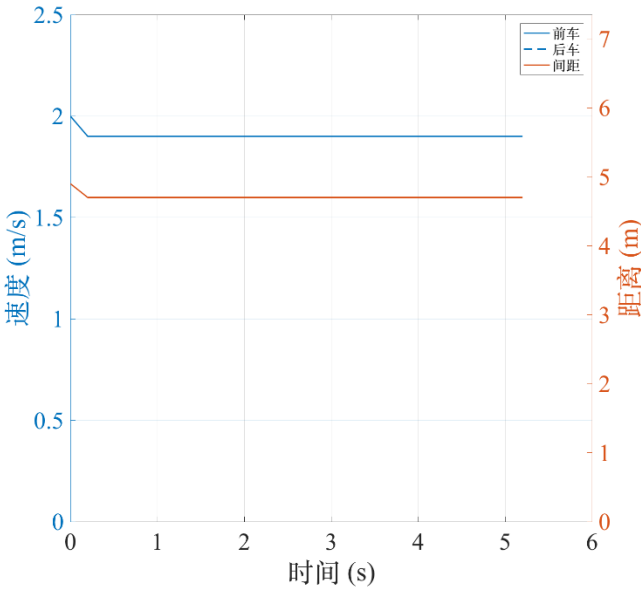


图 4.5 “无人车+普通车”情况 1 仿真结果

我们将前车和后车的初速度均指定为 2 m/s，进行仿真，结果如图 4.5 中的双轴折线图所示。

分析图 4.5，可见前车和后车的速度均从最开始的 2 m/s 进一步下降，最终稳定在 1.9 m/s 左右，并在此后保持速度和距离的稳定。

分析该协同过程，两车最终未达到最优速度的可能原因在于，由于计算能力有限，我们的训练还不够充分完善，故前车在开始时出现速度下降的非理性情况。而我们分析的场景是“无人车+普通车”，普通车采取线性跟驰模型实现对无人车的跟驰，故普通车也出现速度下降的情况，进而使得两车均稳定在较低速度。

##### 2) 前车初速度较小、后车初速度较大

该种情况可被视作后车减速、前车加速的情景。

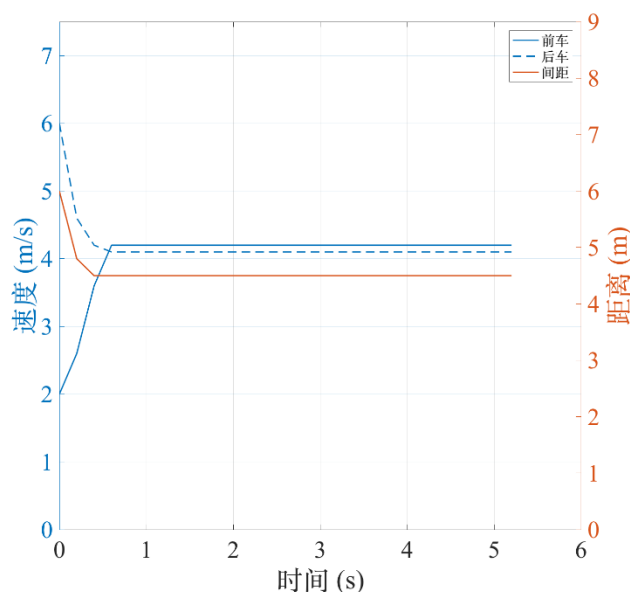


图 4.6 “无人车+普通车”情况 2 仿真结果

我们将前车初速度指定为 2 m/s，后车初速度指定为 6 m/s，进行仿真，结果如图 4.6 中的双轴折线图所示。

分析图 4.6, 可见前车速度从最开始的 2 m/s 较为迅速地增大到 4.2 m/s 左右，而后车速度则从最开始的 6 m/s 下降到 4.1 m/s 左右。之后，两车均保持速度与距离的稳定。值得注意的是，最终稳定时，两车速度并不完全相等。原因在于，前后车速度差造成的距离变化小于离散化的精度范围，使得距离维持不变，进而导致该状态被认定为稳定状态。但在实际应用中，我们可以加以改进，将前后车的速度设定为完全相同。

分析该协同过程，两车的加速、减速过程均较为理想，但最终仍未达到最优速度。可能原因在于，我们分析的场景是“无人车+普通车”，普通车采取线性跟驰模型实现对无人车的跟驰，而开始时无人车车速小于普通车车速，故普通车车速下降以实现跟驰过程。而两车初始速度的均值为 4 m/s，故两车最终也稳定在该速度附近。

### 3) 前车初速度较大、后车初速度较小

该种情况可被视作后车加速、前车减速的追及过程。

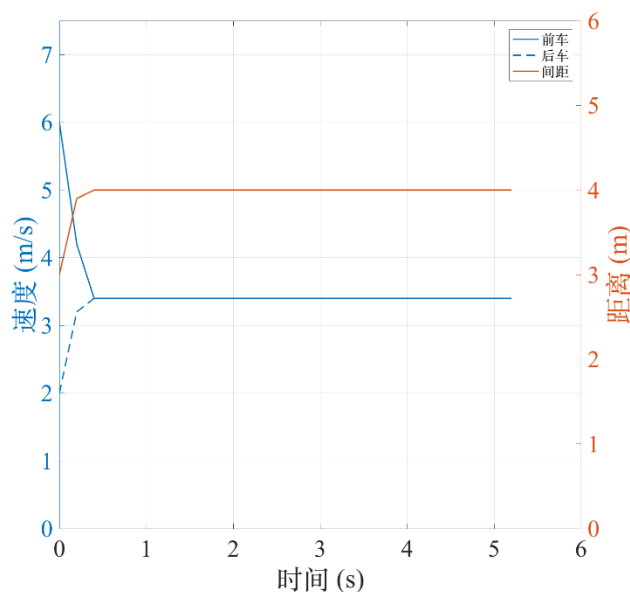


图 4.7 “无人车+普通车”情况 3 仿真结果

我们将前车初速度指定为 6 m/s，后车初速度指定为 2 m/s，进行仿真，结果如图 4.7 中的双轴折线图所示。

分析图 4.7，可见前车速度从最开始的 6 m/s 较为迅速地减小到 3.4 m/s 左右，而后车速度则从最开始的 2 m/s 迅速增加到 3.4 m/s 左右。之后，两车均保持速度与距离的稳定。

分析该协同过程，两车的加速、减速过程均较为理想，但最终仍未达到最优速度。可能原因在于，我们分析的场景是“无人车+普通车”，普通车采取线性跟驰模型实现对无人车的跟驰，而开始时无人车车速大于普通车车速，故普通车车速上升以实现跟驰过程。在这一过程中，两车速度均达到 3.4 m/s，故两车最终也稳定在该速度附近。

#### 4) 前车初速度较大、后车初速度较大

该种情况可被视作两辆车的同步减速情景。

我们将前车和后车的初速度均指定为 6 m/s，进行仿真，结果如图 4.8 中的双轴折线图所示。

分析图 4.8，可见前车速度则出现了不合理的“波折”现象：从最开始的 6m/s 下降到最优速度以下，再恢复到最优速度 5 m/s；后车速度则较为平缓地下降到我们设定的最优速度 5 m/s。在达到最优速度后，两车均保持速度与距离的稳定。



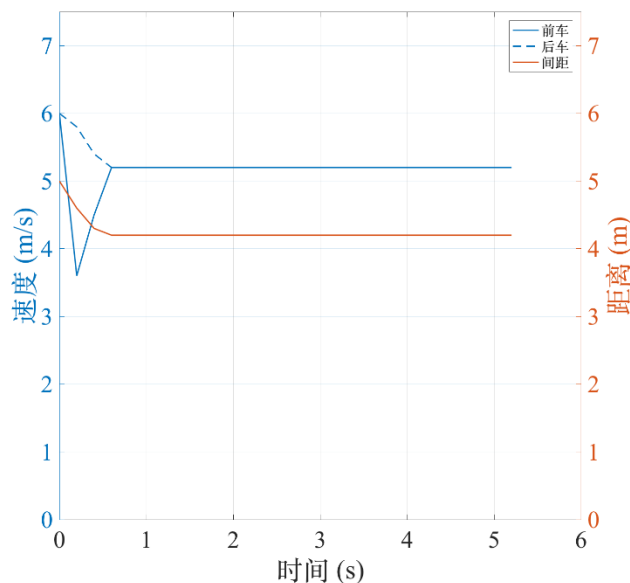


图 4.8 “无人车+普通车”情况 4 仿真结果

分析该协同过程，前车在开始时出现曲线波折的原因可能在于，而我们分析的场景是“无人车+普通车”，普通车采取线性跟驰模型实现对无人车的跟驰，只有前车与后车之间出现速度差时，后车才会相应地通过跟驰策略进行加速/减速。在本场景下，在开始时前车速度的急剧下降目的即是为了制造较大的速度差，诱使后车采用跟驰策略进行减速。而当后车已经减速到了最优速度时，前车将速度提升，以遏制后车的进一步减速，从而实现两车均以最优速度行驶。

## 4.7 性能分析

以下将从覆盖率和训练耗时两方面对 Q-算法的在“无人车+无人车”和“无人车+普通车”场景下的性能进行分析。

### 4.7.1 覆盖率

我们将“覆盖率”定义为：在训练过程中，智能体访问过的状态空间数目在总状态空间数目中的比例。

在本项目中，由于问题的固有性质，状态空间的大小比较大，难以通过训练将其完全覆盖。故在分析 Q-学习算法的算法性能时，在有限的训练次数内能探索并覆盖的状态空间的数目将成为十分关键的指标。

#### 1) 无人车+无人车

在“无人车+无人车”的情况下，我们在 Q-学习算法进行训练时采集相关信息并将状态空间覆盖率与训练次数之间的关系绘制如图 4.9 所示。

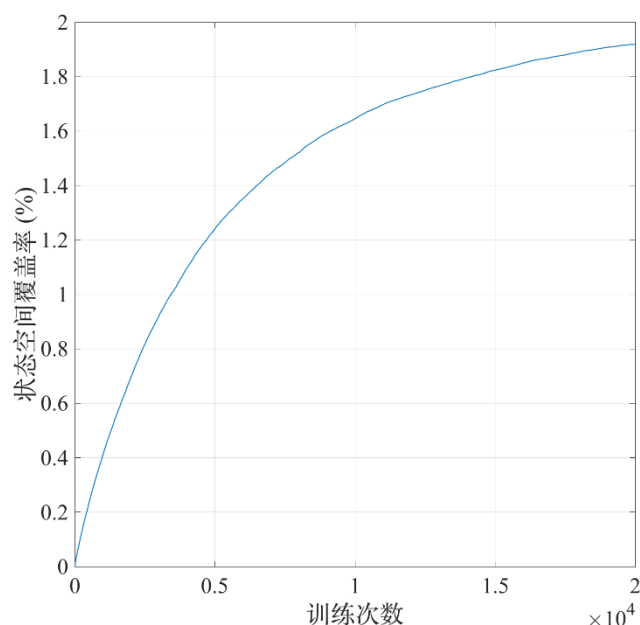


图 4.9 “无人车+无人车”状态空间覆盖率与训练次数关系曲线图

可见，在训练开始时，覆盖率随训练次数的增加而快速增长，但随着训练次数的增加，覆盖率的增长速度也逐渐放缓。在训练次数达到 20000 次的时候，覆盖率在 1.9% 左右。可能原因在于，一方面，根据式(4-4)可知，随着训练次数的增加， $\epsilon$  单调减小，使得智能体探索此前未访问过的状态空间的可能性降低；另一方面，随着训练逐渐趋于充分，最优多车协同策略基本得到确定，训练的随机性进一步减小。

## 2) 无人车+普通车

在“无人车+普通车”的情况下，我们在 Q-学习算法进行训练时采集相关信息并将状态空间覆盖率与训练次数之间的关系绘制如图 4.10 所示。

对比图 4.9 和图 4.10，我们能发现，“无人车+普通车”的情况与“无人车+无人车”十分相似：训练开始时，覆盖率随训练次数的增加而快速增长，但随着训练次数的增加，覆盖率的增长速度也逐渐放缓。在训练次数达到 20000 次的时候，覆盖率在 1.9% 左右。

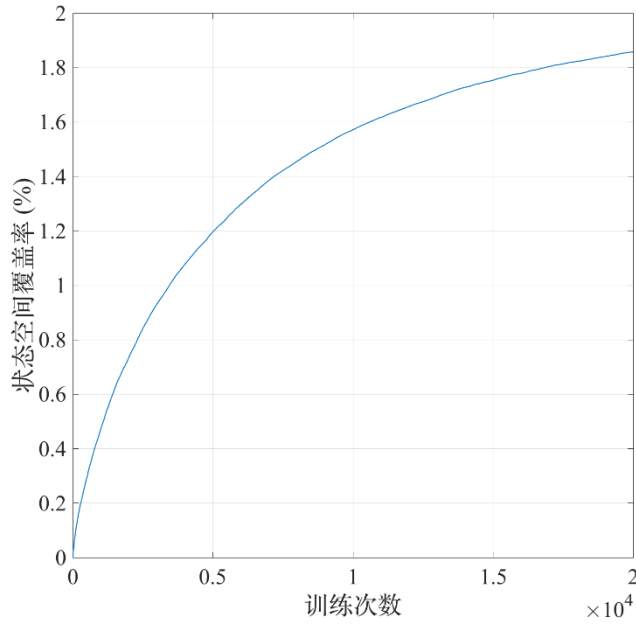


图 4.10 “无人车+普通车”状态空间覆盖率与训练次数关系曲线图

综合对比“无人车+无人车”与“无人车+普通车”的覆盖率增长情况，我们能发现，在优化多车协同策略时，为使状态空间覆盖率收敛所需的训练次数较多。考虑到计算性能与时间的限制，我们最终将训练次数取为  $N = 20000$ ，依旧能得到较为不错的训练效果。

#### 4.7.2 训练耗时

衡量运算性能的另一项指标是训练耗时情况。本项目的训练过程在 MacBook Pro 2016 上进行，训练耗时测算的环境为：

- (1) 中央处理器：2.7 GHz Intel Core i7；
- (2) 内存：16 GB 2133 MHz LPDDR3；
- (3) 训练环境：MATLAB for Mac R2016b。

我们在“无人车+无人车”和“无人车+普通车”场景下，分别进行记录与分析讨论。

##### 1) 无人车+无人车

在“无人车+无人车”的情况下，我们在训练时记录相关数据，并将将单次训练平均耗时与训练次数之间的关系曲线进行绘制，如图 4.11 所示。

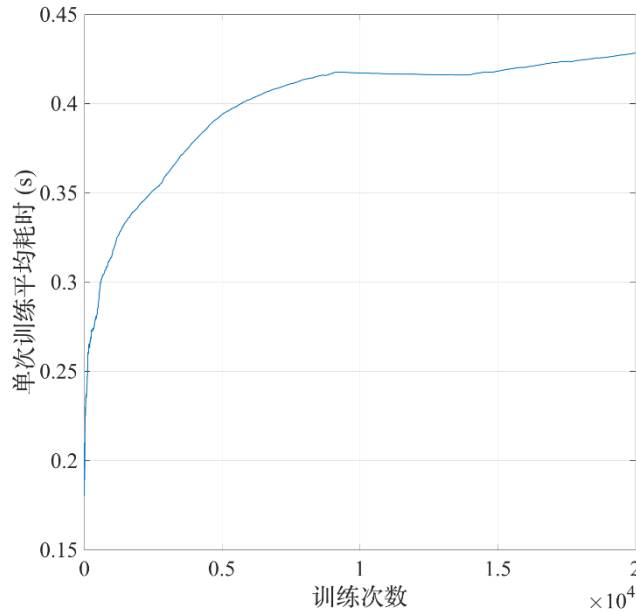


图 4.11 “无人车+无人车”单次训练平均耗时与训练次数关系曲线图

可见，单次训练的平均耗时随着训练次数的增加呈现先急速增加后趋于稳定的趋势。

可能原因为，在起始时，由于状态空间基本未得到训练，两车协同状况比较容易处在极端情形，下一时刻的状态空间大小较为有限；而在训练中后期，多车协同策略已经被训练得较为成熟，两车协同状况更为保守，下一时刻可能的状态空间较大，故每一步进行状态转移时的搜索时间也会增加。

## 2) 无人车+普通车

在“无人车+普通车”的情况下，我们在训练时记录相关数据，并将将单次训练平均耗时与训练次数之间的关系曲线进行绘制，如图 4.12 所示。

可见，与“无人车+无人车”的情况类似，“无人车+普通车”情况下的单次训练的平均耗时随着训练次数的增加也呈现先急速增加后增速放缓的趋势。但在增速放缓后，可明确观察到，训练的平均耗时依旧呈现出线性增加趋势，只是趋势比较平缓。

可能原因已经在“无人车+无人车”的状况中进行分析，此处不再重复。

值得一提的是，对比图 4.11 和图 4.12，可以明确地发现，“无人车+普通车”的情况下，单次训练平均耗时远远小于“无人车+无人车”情况的耗时，前者约为后者的十分之一。

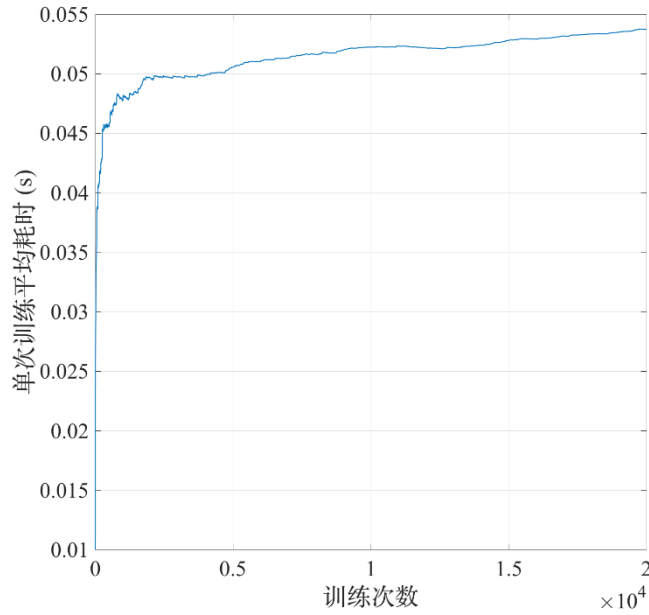


图 4.12 “无人车+普通车”单次训练平均耗时与训练次数关系曲线图

而出现该现象的原因也比较清晰：在“无人车+普通车”的情况下，普通车在下一时刻的行驶状况是可由线性跟驰模型所确定的，不需要像“无人车+无人车”的情况那样多进行一轮搜索，故状态转移部分的耗时显著减少，进而使得训练平均耗时大大降低。

## 4.8 本章小结

在本章，我们将 Q-学习算法应用到多辆车之间的协同问题上。除了在原理、参数设定、状态空间设计、策略集设计和奖赏函数设计等理论层面说明 Q-学习算法的应用之外，我们还进行了仿真实验，对仿真得到的优化结果和仿真算法本身的性能进行了详尽分析。

在理论方面，我们先说明了如何将两车协同策略的优化问题转化为单智能体的增强学习问题。接着，从应用场景的性质出发，参数设定方面对学习速率因子、奖励折现因子、训练次数、贪婪参数和状态转移时间进行了设定与调整。之后，在状态空间设计方面，我们针对状态空间参数选择、范围选择和离散化精度选择的问题进行了详细的分析与论证。最后，我们设计了奖赏函数，并详细说明了设计理由。

在仿真方面，我们分别针对“无人车+无人车”和“无人车+普通车”的场景进行训练，并在各自场景下对多种典型初始状况进行仿真测试，收获了较为良好的效果。此外，我们利用在仿真过程中记录的性能数据，说明了算法在实际应用中的可行性。

## 第5章 信号灯控制策略优化

信号灯控制问题是非常典型的参数优化问题。考虑到交叉口的模型极其复杂，难以用传统的解析方法进行分析与优化，而智能优化算法非常适用于解决参数优化问题，考虑到仿真平台的仿真功能与现有较为成熟的智能优化算法体系，我们将采用智能优化算法来实现信号灯控制策略的优化。

### 5.1 模型设计

#### 5.1.1 问题构建

如 2.3 节所述，我们可用四个参数刻画信号灯的控制策略：前三个相位各自在周期中所占的比例以及信号灯整体相位延时。信号灯的控制参数说明如表 5-1 所示。

表 5-1 信号灯控制策略参数

描述	符号
图 2.2 a) 相位时长所占周期比例	$p_1$
图 2.2 b) 相位时长所占周期比例	$p_2$
图 2.2 c) 相位时长所占周期比例	$p_3$
相延时间	$t_d$

根据表 5-1 的设定，我们可采用仿真的方法，仿真计算出在信号灯作用下，通过交叉口的平均时间。可用函数表示如下：

$$T(p_1, p_2, p_3, t_d) \tag{5-1}$$

由于在研究信号灯配时问题时，我们的目的是使得在信号灯作用下，车辆通过交叉口的时间尽可能短，故目标应为最小化式(5-1)。再考虑到实际应用限制，每一相位在周期中所占的时间比例都应为正数。而时间比例的和应为 1，在此约

束条件的作用下， $p_4$  可用  $p_1$ 、 $p_2$  和  $p_3$  表示。综合以上考虑，信号灯配时问题可用数学语言表述为

$$\begin{aligned} \min_{\{p_1, p_2, p_3, t_d\}} & T(p_1, p_2, p_3, t_d) \\ \text{s. t. } & p_i > 0, i = 1, 2, 3 \\ & \sum_{i=1}^3 p_i < 1 \end{aligned} \quad (5-2)$$

在完成以上建模后，交叉口的信号灯配时问题可被视作典型的带约束条件的非线性优化问题。

### 5.1.2 优化方案

在随机优化领域，模拟退火算法、遗传算法和粒子群算法三种算法均为经典的智能优化方法。这些算法在各个领域均拥有广泛的应用，普适性较强，相关资料也较为全面。

在本项研究中，我们打算对相同的优化问题分别应用模拟退火算法、遗传算法和粒子群算法进行信号灯配时优化，并对优化结果和算法性能进行对比分析，从中分析并选择出最适合本问题的方法。

为了在对比算法性能时更为公平，我们规定每种算法的总循环次数在 2000 次左右，而训练环境均为：

- (1) 中央处理器：2.7 GHz Intel Core i7；
- (2) 内存：16 GB 2133 MHz LPDDR3；
- (3) 训练环境：MATLAB for Mac R2016b。

## 5.2 模拟退火算法

模拟退火算法（SA）是一种非确定性、自启发式的智能优化算法，常被用于求解离散型的问题（如旅行商问题）。

### 5.2.1 原理

模拟退火算法的思想源于冶金学中的“退火”方法：加热金属，再通过以适宜的速度使金属冷却，以细化晶粒尺寸、消除金属组织缺陷。



在模拟退火问题中,由于在每一阶段,系统都会有一定的概率接受较差的解,故相较其他算法,模拟退火算法更不容易陷入局部最优解。事实上,模拟退火算法已经被证明能以 1 的概率收敛得到全局最优解,具有相较其他智能优化算法更为优良的特性。

模拟退火算法的计算过程可用流程图表示,如图 5.1 所示。

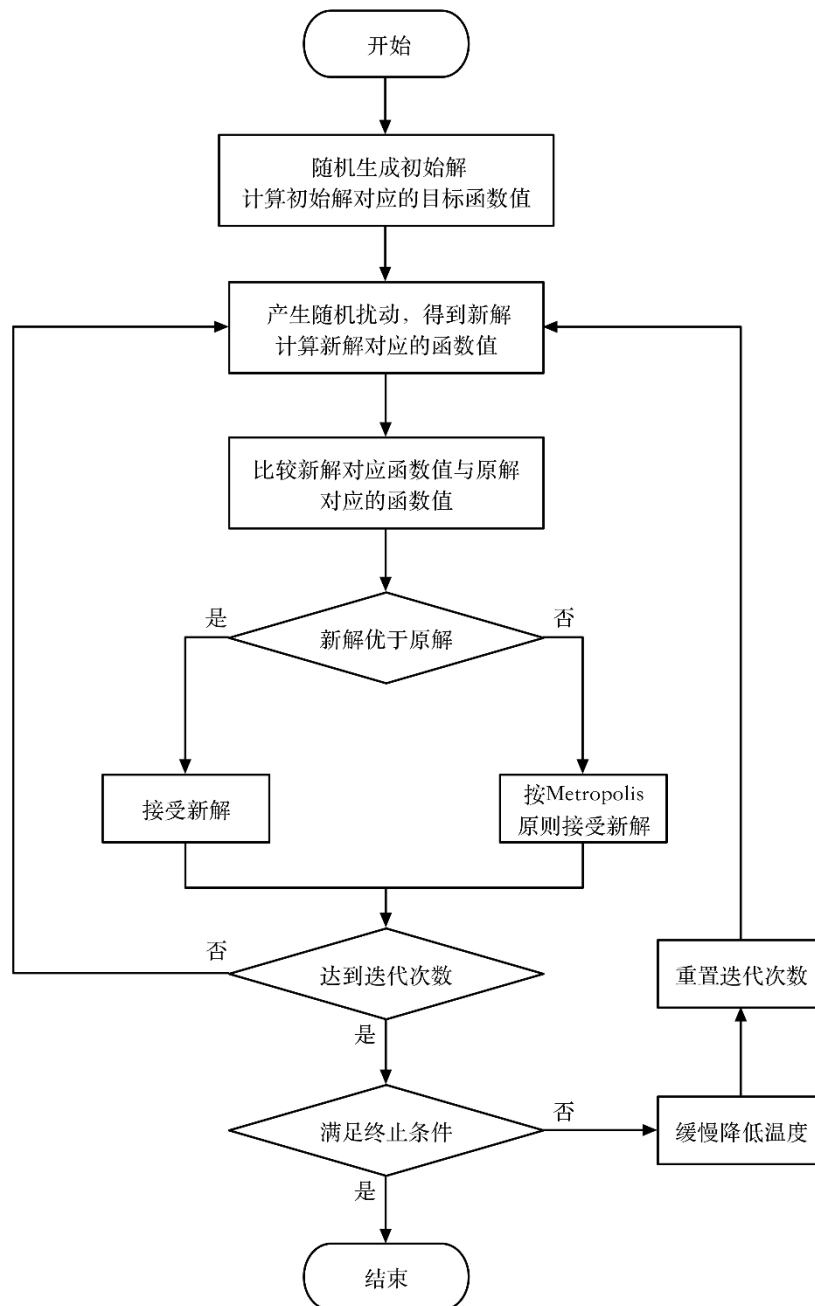


图 5.1 模拟退火算法流程图

算法流程可用文字表述为：

- (1) 初始化充分大的初始温度 $T$ 、每个温度下的迭代次数 $N$ ；
- (2) 随机生成初始解 $S$ （原解）；
- (3) 引入随机扰动，得到新解 $S'$ ；
- (4) 计算  $\Delta T$ ：

$$\Delta T = C(S') - C(S) \quad (5-3)$$

其中 $C(\cdot)$ 为评价函数；

- (5) 若 $\Delta T < 0$ ，将 $S'$ 作为当前解；若 $\Delta T \geq 0$ ，采用 Metropolis 接受准则：

$$\begin{cases} e^{-\frac{\Delta T}{kT}} > r \rightarrow \text{将 } S' \text{ 作为当前解} \\ e^{-\frac{\Delta T}{kT}} \leq r \rightarrow \text{保留 } S \text{ 为当前解} \end{cases} \quad (5-4)$$

其中  $r$  为范围在  $[0,1)$  的随机数；

- (6) 判断在该温度下是否已达到迭代次数，若是，进入步骤 (7)，若否，返回步骤 (3)；
- (7) 判断是否满足终止条件，若是，进入步骤 (8)；若否，降低温度并重置迭代次数，返回步骤 (3)；
- (8) 算法停止，输出结果。

### 5.2.2 参数设定

模拟退火算法中，需要设定的主要参数是初始温度  $T_0$ 、最终温度  $T_1$ 、每个温度下的内循环次数  $L$ 、温度衰减因子  $\delta$  和 Metropolis 接受准则中的参数  $k$  这 5 类。分别设计如下：

#### 1) 初始温度

由于模拟退火算法的参数较多，故我们在设定初始温度时，自由度较大，不妨设定  $T_0 = 100$ 。

#### 2) 最终温度

根据 Metropolis 接受准则，温度在初始温度和最终温度时，采纳较劣解的概率存在如下关系

$$p_1 = e^{-\frac{\Delta T}{kT_1}} = e^{-\frac{\Delta T}{kT_0} \frac{T_0}{T_1}} = p_0^{\frac{T_0}{T_1}} \quad (5-5)$$

根据式(5-5)中  $p_0$  与  $p_1$  之间的幂关系,再考虑到  $0 < p_1 < p_0 < 1$ ,我们认为  $T_0/T_1 = 10$  是较为合适的取值,能使得  $p_1$  相比  $p_0$  有足够的衰减,但又不至于陷入局部最优。此时  $T_1 = 10$ 。

### 3) 内循环次数

在设定初始温度和终止温度后,我们设定每个温度下的内循环次数。考虑到模拟退火算法的核心在于降温过程和以概率接受较劣解,故内循环次数不必太多。在此处,设定  $L = 10$  即可。

### 4) 温度衰减因子

如 5.1.2 节所述,循环次数要设定在 2000 左右,故我们可列写关系式

$$L \cdot \log_{\delta} \frac{T_1}{T_0} \approx 2000 \quad (5-6)$$

我们可根据式(5-6)改写为

$$\delta \approx \left(\frac{T_1}{T_0}\right)^{\frac{L}{2000}} = \left(\frac{100}{10}\right)^{\frac{10}{2000}} \approx 0.9886 \approx 0.99 \quad (5-7)$$

### 5) Metropolis 接受准则中的参数 $k$

在设定 Metropolis 接受准则中的参数  $k$  时,根据式(5-4)的定义,当  $k$  较大时,算法越容易接受较劣解,对周围空间的探索也越充分。经过反复试验,最终发现  $k = 100$  的效果较好。

## 5.2.3 优化结果

按照 5.2.2 节的参数设定,我们进行仿真。交叉口通行时间的优化结果随外循环迭代次数的变化关系曲线如图 5.2 所示。

在图 5.2 中,可见在初始时,交叉口通行时间在 34 s 左右。随着外循环迭代过程开始,最优交叉口通行时间剧烈震荡,同时均值也下降到了 31.5 s 左右。在

外循环迭代次数达到 140 次左右时，最优交叉口通行时间在 29.5807 s 达到稳定，之后曲线进入平稳状态。

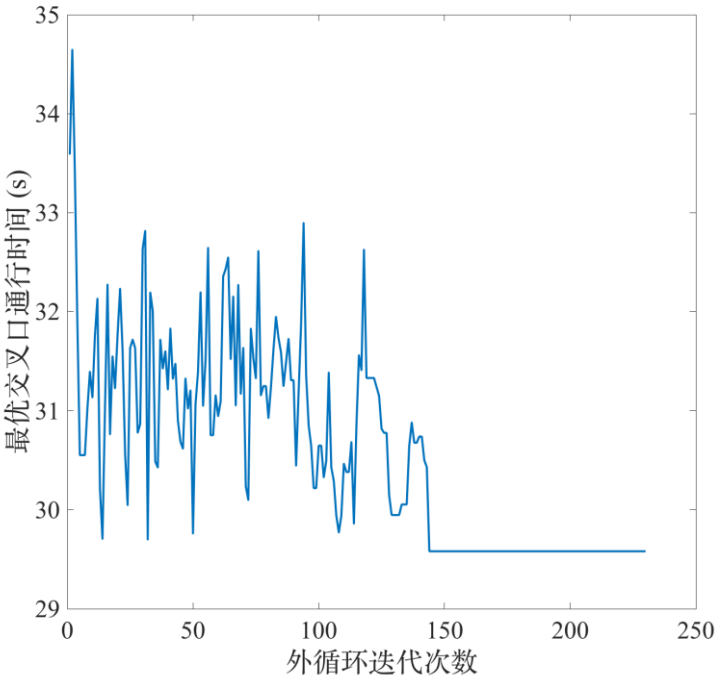


图 5.2 模拟退火算法优化结果

分析模拟退火算法的优化过程，我们发现图 5.2 完全符合模拟退火算法的机理：在外循环迭代前期阶段，温度较高，智能体接受较劣解的概率较高，故最优交叉口通行时间的震荡较为剧烈；而随着外循环迭代次数的增加，温度越降越低，智能体接受较劣解的概率也越来越低；当外循环迭代次数足够多时，智能体接受较劣解的概率已经相当低，而另一方面，此时最优交叉口通行时间也较小，难以优化出更优结果，故交叉口通行时间的优化结果随外循环迭代次数的变化关系曲线进入平缓阶段。

#### 5.2.4 性能分析

我们按照 5.1.2 节中的环境进行仿真计算，并在小循环内的每一次迭代均记录了计算耗时，以分析算法性能以及其所耗时间的变化趋势，以对算法的计算效能进行评价。

我们将结果在图 5.3 中以三维曲面的形式予以绘制。

分析图 5.3，可发现，在绝大多数的仿真内，计算耗时都不超过 2 s，使得充分训练的时间代价不会特别大。

此外，在内循环的内部，计算耗时与内循环迭代次数之间的相关性较小，存在一定的震荡；而在外循环过程中，计算耗时随外循环次数的增加会产生较为剧烈的震荡，但总体而言，计算耗时的平均值随外循环次数的增加有一定的增长。

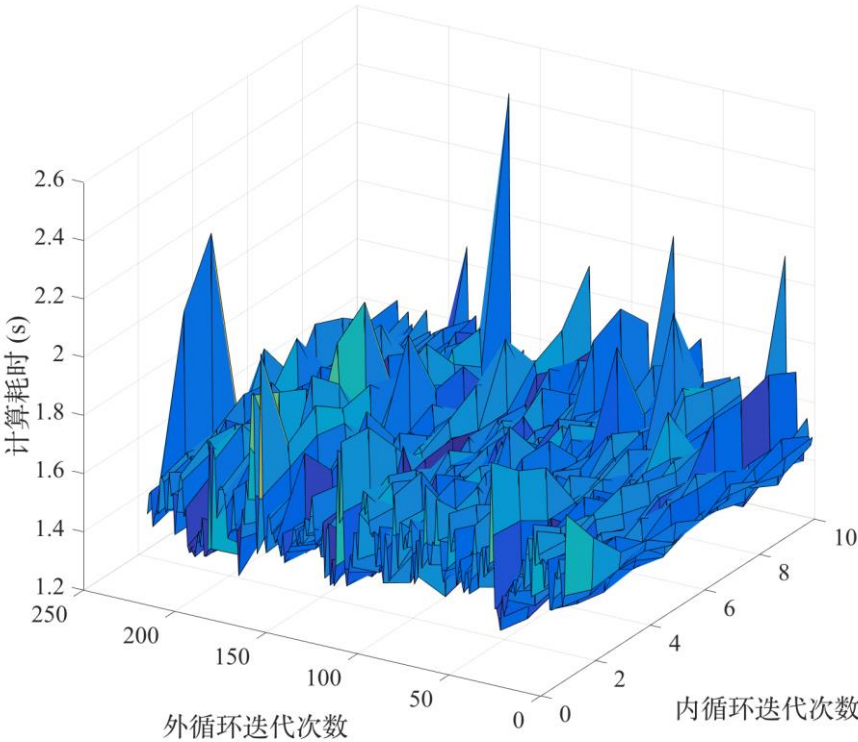


图 5.3 模拟退火算法计算耗时

可能的原因在于，在内循环迭代过程的内部，参数并未发生明显变化，故运算过程与迭代次数之间不存在明显关系，震荡过程只是自然涨落现象；而在外循环迭代的过程中，随着迭代次数的增加，训练越来越充分，信号灯配时方案也越来越优，此时在交叉口的排队现象会有所缓解，交叉口的通行能力上升，更多的车辆能在交叉口通行，在一定程度上增大了计算压力，故计算耗时也会相应地增加。

### 5.3 遗传算法

遗传算法（GA）是一种自启发式的算法，是更为广义的演化算法中的一种，被广泛应用在复杂系统优化与运筹学相关研究当中。

### 5.3.1 原理

其主要受自然界中物种遗传变异机制的启发，通过模仿突变、交叉、优选等生物遗传过程，进而得到在优化、搜索问题中的较优解。

遗传算法种类繁多，其中较有代表性的一种思路如图 5.4 所示。

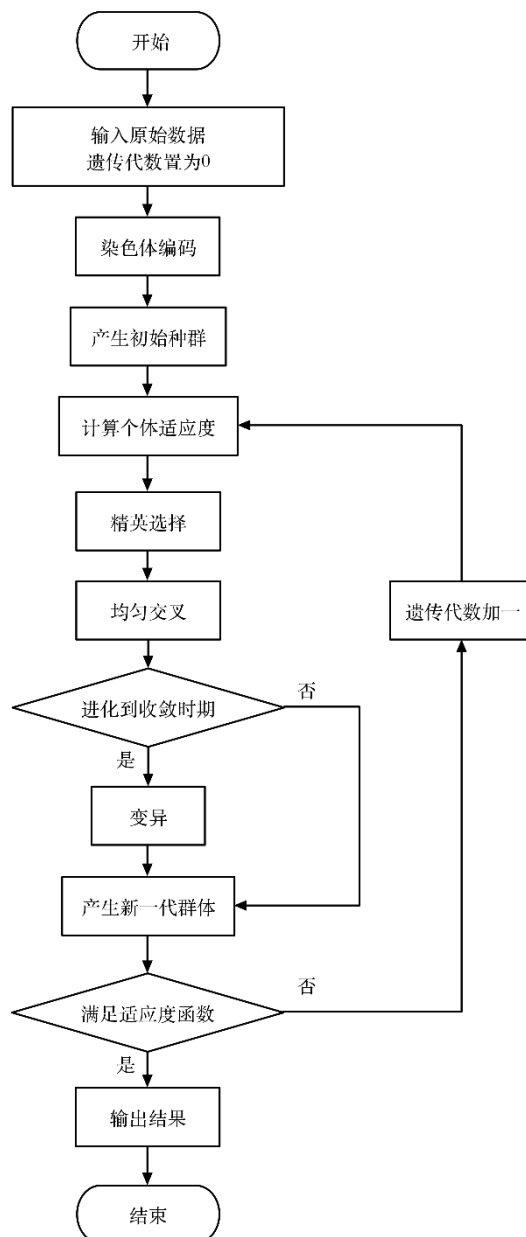


图 5.4 遗传算法流程图

图 5.4 表示的算法具体描述为：

- (1) 初始化最大遗传代数  $T$ ，进行染色体编码，随机生成初始种群；
- (2) 计算群体中各个个体的适应度（评价函数）；
- (3) 选取适应度较好的个体，进行交叉（使用交叉算子作用）；
- (4) 判断是否已进化到收敛时期，若是，进行变异（使用变异算子作用），进入步骤 (5)，若否，直接进入步骤 (5)；
- (5) 产生新一代群体；
- (6) 判断是否满足终止条件，若是，进入步骤 (7)，若否，遗传代数增加一并重置迭代次数，返回步骤 (2)；
- (7) 算法停止，输出结果。

### 5.3.2 算法应用

由于遗传算法更偏向于是一种思想，本身框架比较笼统简略，相关的算法细节需要具体问题具体分析。具体到本问题，我们需要在编码、适应度函数、遗传过程、交叉方法和变异方法方面进行设计。分别设计如下：

#### 1) 编码

由于信号灯的配时方案的参数均是连续量，若要应用传统遗传算法，需要将配饰方案的参数离散化之后再进行编码，但如此一来，一方面会大大增加计算量，另一方面在离散化编码后，参数的精度会有所损失。考虑到遗传算法有连续编码的解决方案，故我们采用连续编码的方式对本问题进行编码。具体而言，我们直接将待优化的参数作为连续编码即可。在连续编码的情境下，交叉方法和变异方法均需要单独设置。

#### 2) 适应度函数

适应度函数起到的重要作用之一是构成选取进行交叉操作的个体的依据，故我们需要先分析交叉操作个体的选取过程。一般而言，参数越优时，适应度函数应越大。在该情境下，一个通行的做法是采用轮盘赌的方式进行交叉个体选择。具体而言，假设适应度函数为  $f(\cdot)$ ，个体总数为  $N$ ，个体  $i$  的参数为  $X_i$ ，则个体  $i$  被选中的概率为

$$p_i = \frac{f(X_i)}{\sum_{j=1}^N f(X_j)} \quad (5-8)$$

假设参数为  $X_i$  时，交叉口通行时间为  $T(X_i)$ 。由于在本项目中，我们的优化目标是尽可能减少交叉口通行时间，故适应度函数应随交叉口通行时间的减小而增大。

我们曾考虑过  $f(X_i) = 1/T(X_i)$ ，但由于  $T(X_i)$  随  $X_i$  变化而产生的相对变化较小（仅有不到 10%），故在该种设计方案下，不论适应度函数如何，各个体被选中的概率相差极小，近似于均匀分布，不符合筛选机制的要求。

为此，我们希望能构造函数，使得  $T(X_i)$  即使只有轻微的减小， $f(X_i)$  都能显著地增大。为此，我们考虑采用指数函数，即  $f(X_i) = \exp(-T(X_i))$ 。但如此一来，出现的问题是，由于  $T(X_i) \approx 30$ ，进行指数运算后，数值过小，不利于进行浮点数运算。

于是我们进一步进行改进。考虑到

$$\frac{e^{-T(X_i)}}{\sum_{j=1}^N e^{-T(X_j)}} = \frac{e^{T_0} \cdot e^{-T(X_i)}}{e^{T_0} \cdot \sum_{j=1}^N e^{-T(X_j)}} = \frac{e^{T_0 - T(X_i)}}{\sum_{j=1}^N e^{T_0 - T(X_j)}} \quad (5-9)$$

我们可采用  $f(X_i) = \exp(T_0 - T(X_i))$  的形式，使得  $f(X_i)$  的范围适中，便于计算，同时也不改变各个体被选择的概率。考虑到  $T(X_i) \approx 30$ ，我们令  $T_0 = 30$ ，即采用

$$f(X_i) = e^{30 - T(X_i)} \quad (5-10)$$

作为适应度函数。最终取得了较为理想的结果。

### 3) 遗传过程

在传统的遗传算法中，理论上新一代个体应全由上一代个体交叉产生。如此一来，增加了对空间的搜索范围，但另一方面，上一代个体训练得到的最优结果难以得到遗传，训练过程中，训练结果的质量可能会下滑。为此，我们在此处应用“精英主义”的理念，即在遗传过程中，将上一代个体中最优秀的个体保留到新一代。具体而言，我们在实现遗传过程时，采用的方式是：先将由上一代个体交叉产生新一代的全部个体，然后再用上一代中的最优个体替代新一代中的最劣个体，较为简洁地实现了“精英选择”机制。

### 4) 交叉方法



由于我们采用的是连续编码的方式，故交叉方法相比传统的遗传算法也有所不同。具体而言，假设上一代被选中的个体的参数分别为  $X_i$  和  $X_j$ ，那么新一代对应的个体的参数分别为

$$\begin{cases} X'_i = (1 - \alpha)X_i + \alpha X_j \\ X'_j = \alpha X_i + (1 - \alpha)X_j \end{cases} \quad (5-11)$$

其中  $\alpha$  为交叉常数，取值范围是  $(0, 1]$ 。

#### 5) 变异方法

对于连续编码情况下的遗传算法，传统方法是再引入变异常数后，利用随机数实现参数在最小值与最大值之间的变动。但在本例中，由于参数之间存在约束关系，无法确定各个参数的最小值与最大值。故我们参考模拟退火算法，对各参数引入随机扰动，以实现变异过程。

$$\Delta X = (k_1 r_1, k_2 r_2, k_3 r_3, k_4 r_4) \quad (5-12)$$

其中  $k_i$  ( $i = 1, 2, 3, 4$ ) 为随机扰动的规模； $r_i$  ( $i = 1, 2, 3, 4$ ) 为随机数，取值范围是  $[0, 1]$ 。

### 5.3.3 参数设定

遗传算法中，需要设定的主要参数是遗传代数  $L$ 、种群规模  $N$ 、交叉常数  $\alpha$  和随机扰动规模  $k_i$  这 4 类。分别设定如下：

#### 1) 种群规模

种群规模即为每一代遗传过程中的个体数目，在选择上也需要有所权衡：种群规模过小的话交叉变异不充分，对可能参数空间的探索不够；而种群规模过大的话会严重影响训练速度。经过反复实验，我们发现较好地兼顾到训练效果与训练速度的设定是  $N = 20$ 。

#### 2) 遗传代数

如 5.1.2 节所述，运算次数要设定在 2000 左右，故我们可列写关系式

$$L \cdot N = 2000 \quad (5-13)$$

我们可将式(5-13)改写为

$$L = \frac{2000}{N} = 100 \quad (5-14)$$

### 3) 交叉常数

由于连续编码的情况较为特殊，我们注意到交叉效果与交叉常数之间存在对称性质：若取  $\alpha' = 1 - \alpha$ ，结合式(5-11)，则有

$$\begin{cases} X_i'' = (1 - \alpha')X_i + \alpha'X_j = \alpha X_i + (1 - \alpha)X_j = X_j' \\ X_j'' = \alpha'X_i + (1 - \alpha')X_j = (1 - \alpha)X_i + \alpha X_j = X_i' \end{cases} \quad (5-15)$$

考虑到交叉群体内部，个体的顺序并不会对算法效果产生影响，故我们可以认为，交叉常数为  $\alpha$  时的交叉效果与交叉常数为  $1 - \alpha$  时的交叉效果相同。

因此，我们只需考虑  $\alpha$  的取值范围是  $(0, 0.5]$  的情况。

在交叉常数过小与过大时，交叉效果均不理想。当交叉常数过小时，交叉效果不明显，例如  $\alpha \rightarrow 0$  时，交叉过程几乎不会对参数产生影响；当交叉常数过大时，交叉得到的新一代参数会比较接近，实际效果同样不佳，例如当  $\alpha = 0.5$  时，交叉过程得到的新一代参数均为原有参数的平均值。

最终，经过反复实验，我们发现较好的设定值为  $\alpha = 0.3$ 。

### 4) 随机扰动规模参数

由于  $r_i$  ( $i = 1, 2, 3, 4$ ) 在  $[0, 1]$  上均匀分布，故变异过程中随机扰动的数学期望为

$$E(\Delta X) = \left( \frac{k_1}{2}, \frac{k_2}{2}, \frac{k_3}{2}, \frac{k_4}{2} \right) \quad (5-16)$$

再考虑到  $X$  中，第 1 项元素的数量级在 10 左右，第 2 项~第 4 项元素的数量级在 0.25 左右，而实验发现， $\Delta X$  数量级在  $X$  数量级的 20% 左右训练效果较好。依照该准则进行反复实验，我们发现，随机扰动规模取

$$\begin{cases} k_1 = 5 \\ k_2 = k_3 = k_4 = 0.1 \end{cases} \quad (5-17)$$

时，训练效果较好。

#### 5.3.4 优化结果

按照 5.3.3 节的参数设定，我们进行仿真。交叉口通行时间的优化结果随外循环迭代次数的变化关系曲线如图 5.5 所示。

在图 5.5 中，可见在初始时，交叉口通行时间在 32.2 s 左右。随着外循环迭代过程开始，最优交叉口通行时间呈现阶梯式下降的趋势，且下降过程较为均匀。最终，最优交叉口通行时间在 29.3423 s 达到稳定。

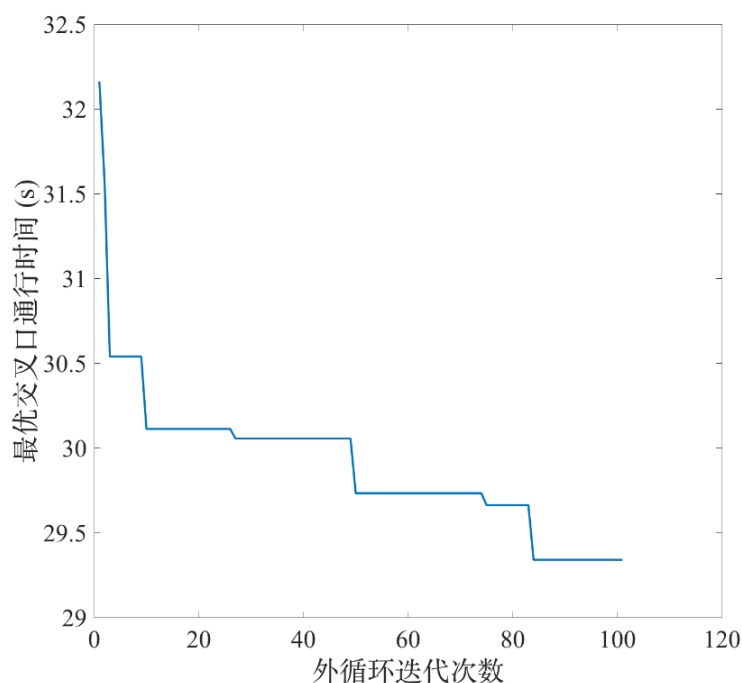


图 5.5 遗传算法优化结果

分析遗传算法的优化过程，我们发现图 5.5 符合我们设计的遗传算法的优化性质。由于我们在遗传算法中引入了“精英主义”，故最优交叉口通行时间随外循环迭代次数是单调不增的，不会出现模拟退火算法中的最优交叉口通行时间剧烈振荡的现象。此外，由于随着遗传代数的增加，遗传算法的交叉、变异机制并不会发生变化，故最优交叉口通行时间的下降较为均匀。

#### 5.3.5 性能分析

我们按照 5.1.2 节中的环境进行仿真计算，并记录了每一代训练的耗时，以分析算法性能以及其所耗时间的变化趋势，以对算法的计算效能进行评价。

结果如图 5.6 中的单次外循环耗时与外循环次数的关系曲线图所示。

分析图 5.6，可发现，在剔除毛刺之后，单次外循环的总耗时随着外循环次数的增加呈现递增趋势，在外循环结束时趋于稳定。考虑到在每次外循环内，在选择交叉的个体时，需要计算各个个体的效用函数，故每次外循环会对 20 个个体各进行一次仿真。平均时间在 1.5 秒左右。

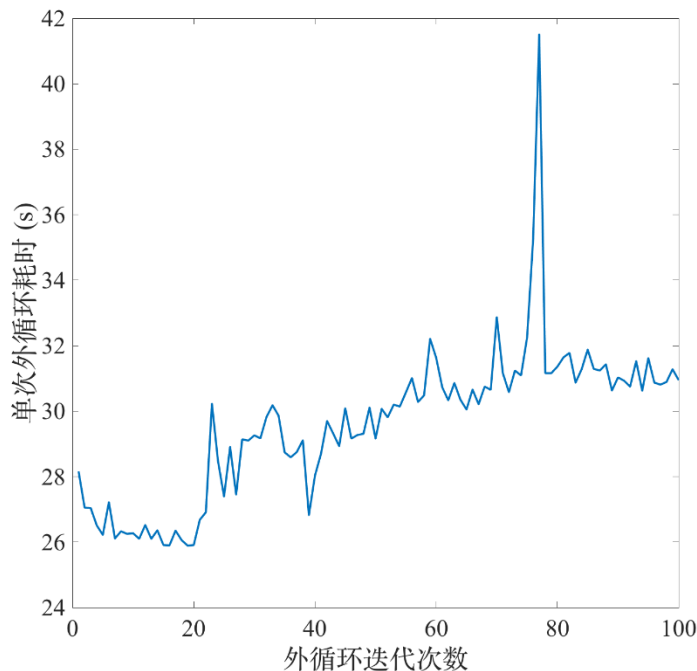


图 5.6 遗传算法计算耗时

分析图 5.6，可能原因在于，在训练开始时，由于信号灯配时方案效果不佳，可能会出现较为严重的排队现象等，导致交叉口通行能力未得到充分利用，交叉口车流量较小，计算荷载较低，故仿真时间较短。而随着外循环迭代次数的增加，信号灯配时方案被不断优化，交叉口的车流量逐渐增大，仿真时的荷载量也逐渐增加，使得仿真耗时有所增加。但最终，最优解逐渐趋于稳定，故仿真耗时也达到了相对稳定。

## 5.4 粒子群算法

粒子群算法（PSO）是一种自启发式的智能优化算法，被广泛使用在函数优化、目标搜索、复杂系统优化、模糊控制等领域。

#### 5.4.1 原理

粒子群算法的思想借鉴了自然界中鸟类觅食的行为，每一粒子都在解空间中游走，其行为会受到自己已有的最佳策略与群体当前所拥有的最佳策略影响，经过不断的游走尝试与信息共享，最优解会逐渐收敛。但值得注意的是，粒子群算法是局部最优算法，所得到的结果有可能只是局部最优。

可用流程图表示，如图 5.7 所示。

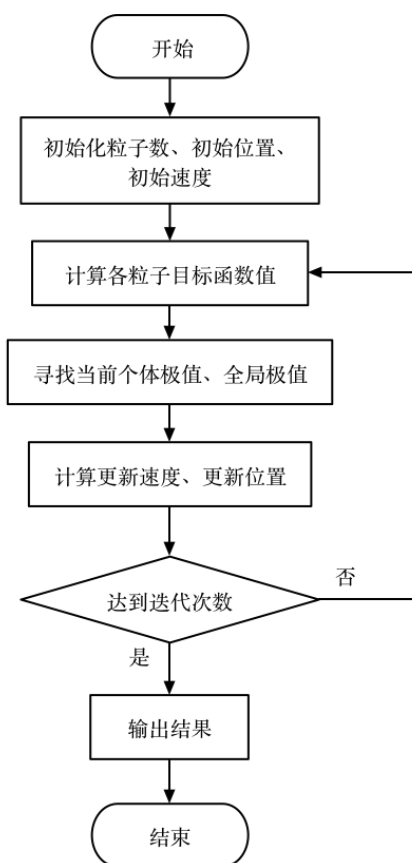


图 5.7 粒子群算法流程图

其步骤可表示如下：

- (1) 初始化粒子数  $N$ 、最大循环次数  $T$ 、初始速度  $V$ 、初始位置  $X$ ；
- (2) 计算各粒子的目标函数值；

- (3) 寻找当前个体的最优位置  $X^*$ ;
- (4) 寻找当前群体的最优位置  $G^*$ ;
- (5) 计算更新速度

$$V \leftarrow w \cdot V + c_1 \cdot r_1 \cdot (X^* - X) + c_2 \cdot r_2 \cdot (G^* - X) \quad (5-18)$$

其中  $w$  为惯性权重,  $c_1$  和  $c_2$  是学习因子,  $r_1$  和  $r_2$  是取值范围为  $[0, 1]$  的随机数;

- (6) 计算更新位置

$$X \leftarrow X + V \quad (5-19)$$

- (7) 判断是否已达到迭代次数, 若是, 进入步骤 (8), 若否, 返回步骤 (2);
- (8) 算法停止, 输出结果。

#### 5.4.2 参数设定

粒子群算法中, 需要设定的主要参数是最大循环次数  $T$ 、粒子数  $N$ 、惯性权重  $w$  和学习因子  $c_i$  这 4 类。分别设定如下:

##### 1) 粒子数

粒子数即为每一次外迭代过程中的个体数目, 在选择上也需要有所权衡: 粒子数过小的话训练不充分, 对可能参数空间的探索不够; 而粒子数过大的话会增大计算量, 在性能上得不偿失。经过反复实验, 我们发现较好地兼顾到训练效果与训练速度的设定是  $N = 20$ 。

##### 2) 最大循环次数

如 5.1.2 节所述, 运算次数要设定在 2000 左右, 故我们可列写关系式

$$T \cdot N = 2000 \quad (5-20)$$

我们可将式(5-20)改写为

$$T = \frac{2000}{N} = 100 \quad (5-21)$$

### 3) 惯性权重

惯性权重反映的是在新一轮迭代中，上一轮迭代的速度参数所占的比重。考虑到经过多次迭代，粒子的速度本身已经蕴含了有关最优解的相关信息，故惯性权重不应过小。经过多次实验，发现  $w = 0.8$  时，优化效果较好。

### 4) 学习因子

学习因子决定了粒子从当前整个群体的最优信息中学习信息的速率与比例。经过多次实验，发现  $c_1 = c_2 = 1$  时，优化效果较好。

## 5.4.3 优化结果

按照 5.3.3 节的参数设定，我们进行仿真。交叉口通行时间的优化结果随外循环迭代次数的变化关系曲线如图 5.8 所示。

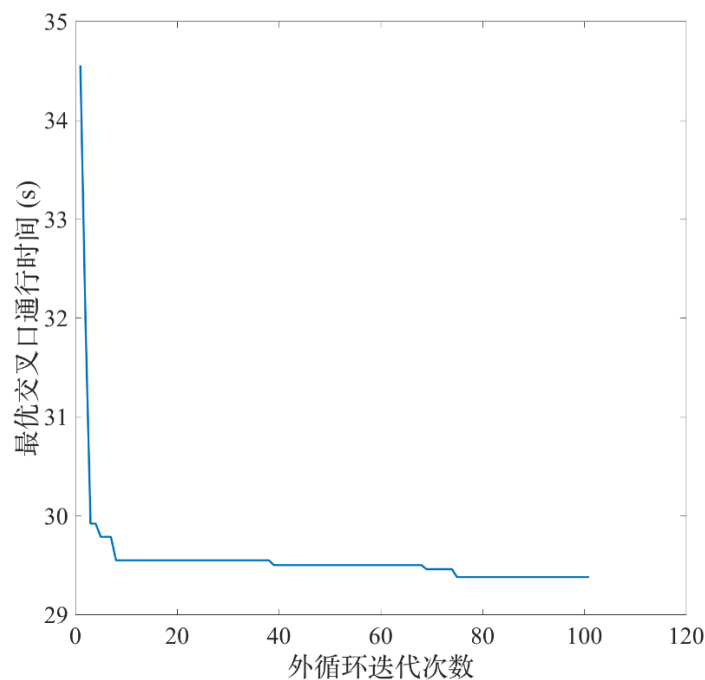


图 5.8 粒子群算法优化结果

在图 5.8 中，可见在初始时，交叉口通行时间在 34.5 s 左右。随着外循环迭代过程开始，最优交叉口通行时间在开始时剧烈下降，在 5 次以内就达到了最终最优解详尽的效果。之后，随着外循环迭代次数的增加，最优交叉口通行时间的

下降过程较为均匀，但变化幅度较小。最终，最优交叉口通行时间在 29.3834 s 达到稳定。

分析粒子群算法的优化过程，我们发现图 5.8 符合粒子群算法的优化性质。由于在粒子群算法中，在每一时刻，当前的最优解都会被记录，且只在有更优解时才会更新。故随着外循环迭代次数的增加，得到的解的质量只会越来越高，最优交叉口通行时间随外循环迭代次数的增加呈现单调不增的趋势。而另一方面，相比遗传算法，粒子群算法在优化过程中，参数的变化方向参考了全局最优信息，故在优化速度较快，在起始时能使最优交叉口通行时间非常陡峭地下降。

#### 5.4.4 性能分析

我们按照 5.1.2 节中的环境进行仿真计算，并在小循环内的每一次迭代均记录了计算耗时，以分析算法性能以及其所耗时间的变化趋势，以对算法的计算效能进行评价。

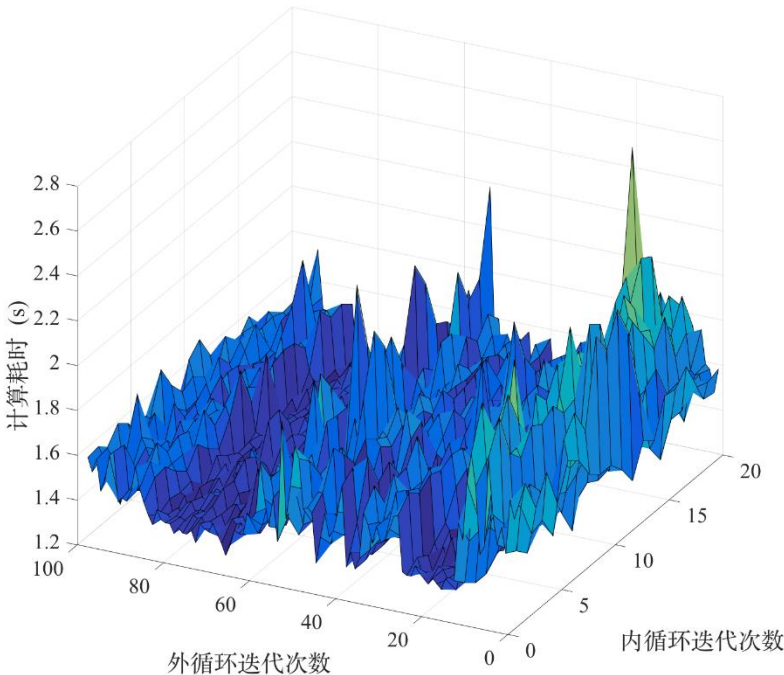


图 5.9 粒子群算法计算耗时

我们将结果在图 5.9 中以三维曲面的形式予以绘制。



分析图 5.9, 可发现, 在绝大多数的仿真内, 计算耗时都不超过 2 s, 使得充分训练的时间代价不会特别大。

分析三维曲面走势, 可以发现, 在内循环内部, 计算耗时呈现出正常涨落的特点, 与内循环次数无关; 在外循环中, 随着外循环迭代次数的增加, 单次外循环的计算耗时的变化不大。

可能的原因在于, 在内循环迭代过程的内部, 参数并未发生明显变化, 故运算过程与迭代次数之间不存在明显关系, 震荡过程只是自然涨落现象。而在外循环迭代的过程中, 在开始时, 根据图 5.8 可知, 优化结果有非常显著而剧烈的提升, 但之后优化结果的提升不大, 最优交叉口通行时间的变化幅度很小。因此, 在此后的外循环迭代过程中, 计算耗时的变化也并不明显。

## 5.5 算法对比

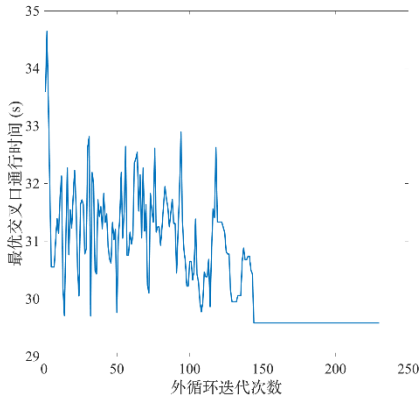
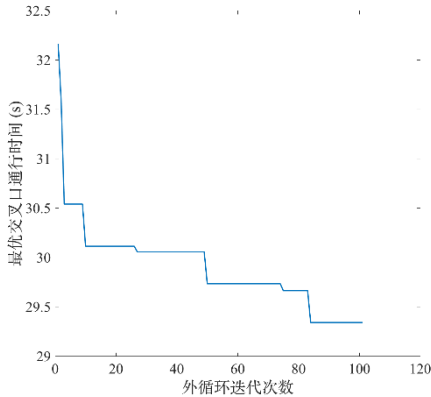
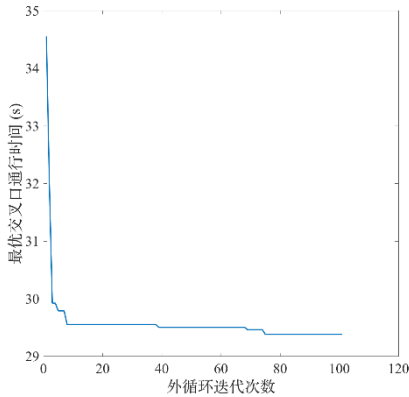
粒子群算法与模拟退火算法、遗传算法均有一定相似之处, 三者均是通过迭代手段从初始解出发搜寻最优解, 也同样采用适应性函数作为评判标准。不过三者之间在性能与具体原理上依旧是有差异的, 如表 5-2 所示。

在最优结果方面, 三种优化方法优化得到的结果较为相近, 均在 29 s ~ 30 s 之间。进一步对比, 可以发现, 模拟退火算法得到的最优交叉口通行时间大于遗传算法和粒子群算法, 而遗传算法和粒子群算法的优化结果十分接近, 可近似认为优化效果相同。

在优化趋势方面, 模拟退火算法在起始时震荡极为剧烈, 但最终找到最优解的速度是最快的; 粒子群算法则在开始时就使得最优交叉口通行时间迅速下降, 很快便得到较优解; 遗传算法介于模拟退火算法与粒子群算法之间, 其优化过程呈现阶梯式现象, 较为均匀。

在性能方面, 我们对比了在三种算法的训练过程中, 单次仿真的平均耗时。注意到, 尽管三种算法采用的是同一套仿真系统, 但由于信号灯配时方案的不同, 会导致交叉口车流量状况的不同, 进而使得计算荷载的情况不同。因此, 不同的算法可能会带来不同的单次仿真平均耗时。三种方法的单次仿真平均耗时较为接近, 且均在 1.5 s 以内。进一步对比, 可以发现粒子群算法在运算速度上相比模拟退火算法和遗传算法略有优势。

表 5-2 智能优化算法对比

算法类型	模拟退火算法	遗传算法	粒子群算法
理论结果	全局最优	局部最优	局部最优
最优结果 (s)	29.5807	29.3423	29.3834
优化趋势			
最终结果 <sup>①</sup>	(2.6409, 0.3135, 0.1037, 0.3949, 0.1879)	(5.2531, 0.4091, 0.0964, 0.3778, 0.1168)	(9.7377, 0.4431, 0.0990, 0.3589, 0.0990)
单次耗时 (s)	1.4943	1.4793	1.4524
最优结果耗时 <sup>②</sup> (s)	2148.97	2429.55	2175.01
较优结果耗时 <sup>③</sup> (s)	2148.97 <sup>④</sup>	1365.47	66.29

① 5 个参数分别代表相延时间和 4 个相位在一个周期中所占的时间比例  
② 从开始训练到最优交叉口通行时间达到最优值的耗时  
③ 从开始训练到最优交叉口通行时间降到 30 s 以下并稳定维持在 30 s 以下的耗时  
④ 最优交叉口通行时间在此时间前也曾降到 30 s 以下，但未稳定存在

在最优结果耗时方面,可发现,模拟退火算法稳定在最优解的速度是最快的;粒子群算法其次,但与模拟退火算法的耗时相差较小;遗传算法收敛到最优解的速度则显著慢于模拟退火算法和粒子群算法。

在实际应用中,30 s 与 29 s 之间的差别并不会很大,而且由于计算能力的限制,我们很难充分训练得到最优解。因此,我们设计了一项指标:较优结果耗时,旨在衡量算法训练得到较为理想的结果时的用时。在此处,我们规定使得交叉口通行时间在 30 s 以下的信号灯配时方案就是较优解。相比“最优结果耗时”,“较优结果耗时”指标衡量的结果则有很大的不同。模拟退火算法的较优结果耗时表现较为糟糕,与最优结果耗时一致;遗传算法的较优结果耗时仅为最优结果耗时的一半左右;而粒子群算法给予了我们很大的惊喜,仅需六十余秒即可得到较优解,具有较强的实用性。

可见,虽然在理论上,模拟退火算法能最终收敛到全局最优解,而遗传算法和粒子群算法只能得到局部最优解。但在本问题中,实验发现,遗传算法和粒子群算法在结果(优化结果、较优结果耗时)和性能(单次耗时)上均取得了优于模拟退火算法的结果。

综合以上结果与分析,我们发现,粒子群算法最优结果与最优结果耗时方面位列第二,但与第一之间的差距十分微小;而粒子群算法在单次耗时与较优结果耗时方面则在三种算法中表现最好。其中,在对实际应用极为重要的“较优结果耗时”方面,粒子群算法具有巨大的优势。因此,我们认为,针对本问题场景下的信号灯配时策略优化,粒子群算法拥有最佳的表现。

## 5.6 本章小结

在本章,我们以交叉口四个信号灯的配时方案为优化对象,在将信号灯配时方案优化问题抽象成带约束的多参数优化问题后,分别采用了模拟退火算法、遗传算法、粒子群算法三种智能优化算法,并对三种算法进行对比。

在采用每一种智能优化算法时,我们都先通过步骤说明与流程图对该方法的原理进行介绍,接着针对不同的算法参数,根据问题性质进行参数分析与设定。我们接着在自行搭建的 MATLAB 仿真平台上利用该算法,以最小化交叉口平均通行时间为目标,对信号灯配时方案进行优化,并分析该算法的性能。

我们还对三种算法进行了详尽的对比。我们在优化结果、达到最优结果的耗时、达到较优结果的耗时等指标方面对三种算法的有效性与可行性进行对比分析,

发现三种算法在最优结果方面差别不大，而粒子群算法具有收敛速度快、鲁棒性强的优点，且粒子群算法得到较优结果的耗时远远小于其它两种算法。因此，我们认为，在该问题场景中，粒子群算法是较为适合进行交叉口信号灯优化的智能优化算法。

## 第6章 总结和展望

我们对项目进行总结，并对未来可能工作进行展望如下。

### 6.1 总结

本项目以无人驾驶与有人驾驶相混合环境下的交叉口为研究对象，以提升交叉口通行能力与通行舒适度为目标，首先完全自主设计并搭建了基于 MATLAB 的仿真测试平台，并设计了针对车辆左转弯轨迹、直线行驶路径上多车协同策略、信号灯配时方案的优化方案。

在仿真测试平台搭建方面，我们以典型的平面十字交叉路口为对象，对其进行抽象化、坐标化处理，并考虑安全因素设计了相应的车道通行策略与信号灯相位。该仿真平台能实现在给定车辆轨迹、协同策略、信号灯配时下进行车辆运动情况的仿真，同时记录历史数据信息以进行对比。此外，本项目还实现了车辆运动过程的动画化，使策略调试与成果展示更为直观。

在车辆左转弯轨迹优化方面，我们采用了 Q-学习的方法，将离散化的车辆位置与行驶方向作为状态空间的组成要素，将行驶速度与舒适度在奖赏函数中予以体现，同时利用问题研究场景本身的性质减少了状态空间的大小。最终训练出的结果实现了速度与舒适度之间的权衡，且经验证，最优轨迹是现实可行的。

在直线行驶路径上的多车协同策略优化方面，我们通过将车辆两两组队，以两辆车为整体应用 Q-学习算法。我们令离散化的两车速度与两车间距构成状态空间，同时采用反比例函数形式的奖赏函数使得两辆车的速度尽可能接近给定最优速度值。训练得到的策略在各场景下均能提供及时有效的速度控制方案。

在信号灯配时方案优化方面，我们将交叉口信号灯配时优化问题抽象成带约束条件的多参数优化问题，分别采用模拟退火算法、遗传算法、粒子群算法进行优化并对各算法的优化效果与计算性能进行分析。最终，我们得出结论，认为收敛速度快、在获得较优结果的耗时上有显著优势的粒子群算法最为适合本研究场景下的信号灯配时方案优化问题。

### 6.2 展望

本项目主要创新点在于将强化学习算法引入交叉口通行策略优化，通过应用无模型的 Q-学习算法，在非参优化的问题情境中，实现了传统方法难以实现的轨迹优化和车辆协同策略优化。

本项目应用了强化学习这一较为流行的人工智能工具。考虑到当前人工智能浪潮正盛，不同的学习算法之间相互借鉴补充的情况十分普遍，故后续研究可以考虑将更多的人工智能方法引入研究。例如，在本项目中，在采取 Q-学习算法时，需要将连续物理量离散化，一方面使得状态空间极其庞大，影响计算效率与计算效果，另一方面，离散化过程也在一定程度上损害了问题精度。而若将神经网络引入强化学习，可实现对连续状态空间的学习。

## 插图索引

图 1.1 路口结构示意图 .....	5
图 2.1 路口参数 .....	9
图 2.2 交叉口通行模式 .....	11
图 2.3 车辆行驶轨迹示意图 .....	12
图 3.1 增强学习算法流程图 .....	16
图 3.2 状态空间物理含义 .....	20
图 3.3 可行状态空间区域 .....	22
图 3.4 可能状态集合 .....	25
图 3.5 轨迹优化结果图 .....	30
图 3.6 轨迹设计示意图 .....	31
图 3.7 状态空间覆盖率与训练次数关系曲线图 .....	33
图 3.8 单次训练平均耗时与训练次数关系曲线图 .....	34
图 4.1 “无人车+无人车”情况 1 仿真结果 .....	48
图 4.2 “无人车+无人车”情况 2 仿真结果 .....	49
图 4.3 “无人车+无人车”情况 3 仿真结果 .....	50
图 4.4 “无人车+无人车”情况 4 仿真结果 .....	51
图 4.5 “无人车+普通车”情况 1 仿真结果 .....	52
图 4.6 “无人车+普通车”情况 2 仿真结果 .....	53
图 4.7 “无人车+普通车”情况 3 仿真结果 .....	54
图 4.8 “无人车+普通车”情况 4 仿真结果 .....	55
图 4.9 “无人车+无人车”状态空间覆盖率与训练次数关系曲线图 .....	56
图 4.10 “无人车+普通车”状态空间覆盖率与训练次数关系曲线图 .....	57
图 4.11 “无人车+无人车”单次训练平均耗时与训练次数关系曲线图 .....	58
图 4.12 “无人车+普通车”单次训练平均耗时与训练次数关系曲线图 .....	59
图 5.1 模拟退火算法流程图 .....	63
图 5.2 模拟退火算法优化结果 .....	66
图 5.3 模拟退火算法计算耗时 .....	67
图 5.4 遗传算法流程图 .....	68

图 5.5 遗传算法优化结果 .....	73
图 5.6 遗传算法计算耗时 .....	74
图 5.7 粒子群算法流程图 .....	75
图 5.8 粒子群算法优化结果 .....	77
图 5.9 粒子群算法计算耗时 .....	78



表格索引

表 2-1 车辆主要参数 ..... 8

表 2-2 路口结构主要参数 ..... 10

表 3-1 状态空间参数选择 ..... 19

表 3-2 加速度、不适程度、舒适度关系 ..... 28

表 4-1 状态空间参数选择 ..... 41

表 5-1 信号灯控制策略参数 ..... 61

表 5-2 智能优化算法对比 ..... 80

## 参考文献

- [1] Degroot B. Self-Driving Vehicles Generate Enthusiasm, Concerns Worldwide[J]. Umtri Research Review, 2014.
- [2] Urmson C, Whittaker W R. Self-Driving Cars and the Urban Challenge[J]. Intelligent Systems IEEE, 2008, 23(2):66-68.
- [3] Shladover S E. The Truth about “Self-Driving” Cars[J]. Scientific American, 2016.
- [4] Greenblatt N A. Self-driving cars and the law[J]. IEEE Spectrum, 2016, 53(2):46-51.
- [5] Wu Z, Zhao L, Sun J. A Study of Dynamic Right-Turn Signal Control Strategy at Mixed Traffic Flow Intersections[J]. Promet - Traffic - Traffico, 2014, 26(6):449-458.
- [6] Sen S, Head K L. Controlled Optimization of Phases at an Intersection[J]. Transportation Science, 1997, 31(1):5-17.
- [7] Han Q, Zhou Y. Intelligent Control Strategy of Traffic Light at Urban Intersection[J]. International Journal of Online Engineering, 2013, 9(3):111.
- [8] Park B. Enhanced Genetic Algorithm for Signal-Timing Optimization of Oversaturated Intersections[J]. Transportation Research Record, 2000, 1727(1727):32-41.
- [9] Lee J, Park B. Development and Evaluation of a Cooperative Vehicle Intersection Control Algorithm Under the Connected Vehicles Environment[J]. IEEE Transactions on Intelligent Transportation Systems, 2012, 13(1):81-90.
- [10] Nielsen O A, Frederiksen R D, Simonsen N. Using expert system rules to establish data for intersections and turns in road networks[J]. International Transactions in Operational Research, 1998, 5(6):569-581.
- [11] Radwan A E, Benevelli D A. Bus Priority Strategy: Justification and Environmental Aspects[J]. Journal of Transportation Engineering, 1983, 109(1):88-106.
- [12] Sun X. Evaluation of the pre-detective signal priority for bus rapid transit: coordinating the primary and secondary intersections[J]. Transport, 2015:1-11.
- [13] Tan Z B, Yuan H. A Study on Traffic Channelization Design Methods for Complex Intersection[J]. Journal of Highway & Transportation Research & Development, 2006.
- [14] Asaithambi Gowri, Ramaswamy Sivanandan. Evaluation of left turn channelization at a signalized intersection under heterogeneous traffic conditions[J]. Transport, 2010, 99(3):221-229.
- [15] Hou M, Zi-Wei H E, Jia Z X, et al. Coordinated Optimization of Channelization and Signal

- Timing of Intersection Based on Synchro Simulation[J]. Transportation Standardization, 2014.
- [16] Lewis F, Liu D. Reinforcement Learning and Approximate Dynamic Programming for Feedback Control[M]. 2013.
  - [17] Khan S G, Herrmann G, Lewis F L, et al. Reinforcement learning and optimal adaptive control: An overview and implementation examples[J]. Annual Reviews in Control, 2012, 36(1):42-59.
  - [18] Sutton R S, Barto A G, Williams R J. Reinforcement learning is direct adaptive optimal control[J]. IEEE Control Systems, 1991, 12(2):19-22.
  - [19] Werbos P J. Consistency of HDP applied to a simple reinforcement learning problem[J]. Neural Networks, 1990, 3(2):179-189.
  - [20] Barto A G. Connectionist learning for control: an overview[M] Neural networks for control. 1990:5-58.
  - [21] Sutton R S, Barto A G. Reinforcement learning: an introduction[J]. IEEE Transactions on Neural Networks, 1998, 9(5):1054.
  - [22] Werbos P. 1 ADP: Goals, Opportunities and Principles[J]. Introduction to C++ for Financial Engineers: An Object-Oriented Approach, 2004:1-4.
  - [23] Werbos P J. Using ADP to Understand and Replicate Brain Intelligence: The Next Level Design?[J]. Understanding Complex Systems, 2007, 2007:209 - 216.
  - [24] Werbos P J. Foreword - ADP: The Key Direction for Future Research in Intelligent Control and Understanding Brain Intelligence[J]. IEEE Transactions on Cybernetics, 2008, 38(4):898-900.
  - [25] Werbos P J. Intelligence in the brain: A theory of how it works and how to build it[J]. Neural Networks, 2009, 22(3):200-212.
  - [26] Van Roy B, Bertsekas D P, Lee Y, et al. A neuro-dynamic programming approach to retailer inventory management[C] Decision and Control, 1997. Proceedings of the, IEEE Conference on. IEEE, 2000:4052-4057 vol.4.
  - [27] Rudowsky I, Kulyba O, Kunin M, et al. Reinforcement Learning Interfaces for Biomedical Database Systems[C] International Conference of the IEEE Engineering in Medicine & Biology Society. PubMed, 2006:6269-6272.
  - [28] Ernst D, Glavic M, Geurts P, et al. Approximate Value Iteration in the Reinforcement Learning Context. Application to Electrical Power System Control.[J]. International Journal of Emerging Electric Power Systems, 2005, 3(1).
  - [29] Kampen E V, Chu Q P, Mulder J A. Online Adaptive Critic Flight Control using Approximated Plant Dynamics[C] International Conference on Machine Learning and Cybernetics. IEEE, 2006:256-261.
  - [30] Liu W, Tan Y, Qiu Q. Enhanced Q-learning algorithm for dynamic power management with performance constraint[C] Design, Automation & Test in Europe Conference & Exhibition.

IEEE, 2010:602-605.

- [31] Stingu P E, Lewis F L. Adaptive dynamic programming applied to a 6DoF quadrotor[J]. 2011.
- [32] Khan S G, Herrmann G, Lewis F L, et al. A Novel Q-Learning Based Adaptive Optimal Controller Implementation for a Humanoid Robotic Arm \*[J]. IFAC Proceedings Volumes, 2011, 44(1):13528-13533.
- [33] Khan S G, Herrmann G, Lewis F, et al. A Q-learning based Cartesian model reference compliance controller implementation for a humanoid robot arm[C] Robotics, Automation and Mechatronics. IEEE, 2011:214-219.
- [34] Mechanical vibration and shock evaluation of human exposure to whole-body vibration[S]. ISO2631-1:1997.

## 致 谢

首先要感谢我的导师胡坚明老师。我在大一暑假就加入了胡老师的实验室，由此开启了自己的学术道路。在胡老师的实验室里，我第一次发表学术论文，第一次获得科创奖项，第一次参加国际会议。而本次毕业设计，胡老师不论是在题目的选择，还是在研究方案的指导上都给予了我极大的帮助，在此向胡老师表示衷心的感谢！

其次要感谢我的同学们，尤其是郑亦平同学。我在刚开始接触强化学习的概念时，对该领域一无所知，而郑亦平同学给予了我很多帮助。在与郑亦平同学讨论的过程中，我逐渐对强化学习的概念有了初步的了解。而之后，在参数设定等应用实现的细节方面，郑亦平同学也与我分享了自己的经验心得，让我获益良多。此外，郑亦平同学也是我的挚友，在毕业设计不顺的时候给予了我许多鼓励和支持。

此外，我还要感谢实验室里帮助过我的学长学姐，包括张若冰学姐、霍雨森学长、高攀学长、曾镜鸿学长等，我与他们在交通与人工智能方面进行了许多讨论，并从中获益良多。



## 声 明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签 名：\_\_\_\_\_ 日 期：\_\_\_\_\_





## 附录A 外文资料的书面翻译

### 自适应路由选择的强化学习算法

**摘要：**强化学习意味着学习一种策略——从观察到结果的映射——基于从环境中得到的反馈。这个学习过程能被视作搜索一组策略，并通过它们与环境之间的交互结果来进行评价。我们提出了梯度上升算法在增强学习的应用，将其应用于网络通信中数据包选路的复数域，并在一个基准问题上对比了该算法相对其他选路方法的性能。

#### A.1 介绍

成功的电信通信要求有效的资源分配方法，该方法应该通过发展自适应控制策略获得。增强学习（RL）为这些策略提出了一种自然的框架：通过与环境的交互来进行不断的试错。在本项工作中，我们将强化学习算法应用于网络路由。有效的网络路由意味着选择最优通信路径。该问题能被建模成多目标增强学习问题。从某种程度上讲，网络路由的最优控制策略的学习过程能被视作传统的增强学习情景任务，例如迷宫寻路或杠平衡，但相比之下，需要并行进行重复尝试并在尝试之间进行交互。

在这样的理解之下，单个路由器能被视作根据单个个体的策略进行路由规划的智能体。该策略的参数是根据对网络的全局性能的若干度量进行调整的，与此同时控制策略是由局部的观测结果所决定的。节点不具有关于网络的拓扑结构和和自己位置的任何信息。对节点们的初始化和它们所遵循的学习算法在节点之间是相同的，且独立于网络结构。在空间中没有方向的概念，也没有关于行动的含义。我们的方法使得我们能够在更新局部策略的同时，无需集中式控制或网络结构的全局信息。该学习算法所需要的唯一全局信息是网络利用率——表示成在每一跳分配一次且与平均路由时间相关奖励信号。该多目标学习系统拥有生物学上的合理解释，并能被视作神经网络，其中每个神经元仅仅基于局部可行量进行简单计算。

## A.2 域

我们采取了 Boyan 和 Littman 的工作中的域，并在其上测试自己的算法。该域是离散时间情境下的通信网络仿真器，拥有各式各样的拓扑结构和动态结构。通信网络是源于互联网、交通网络等实际系统的抽象表示。它包括一组同质的节点和由节点连接而成的边（如图 1 所示）。彼此相连的节点被称作邻居。连接可能是活跃的（“上”）或不活跃的（“下”）。每个节点都能作为数据包的起点或终点，或者作为路由器。

数据包周期性地进入网络，起点和终点的选择是随机均匀选择的。数据包从起点经过中继节点到达终点。如果起点和终点一致，则没有数据包生成。将一个数据包通过链路进行传输会引发一个代价，可以将其视作传输时间。如果数据包为了获取计算资源而在某些特定节点上进行排队时，会产生附加代价——排队延时。两种类型的代价都被假设成在整个网络中是均匀的。在我们的实验中，每种代价都被设定成单位代价。网络交通的层次是由网络中数据包的数量决定的。一旦数据包到达终点，他就被移除了。如果一个数据包在网络中传输的时间过长，它同样会作为没有希望的事件被移除。多数据包会在节点形成先进先出的有限长度队列。节点必须将其最考上的节点转发给邻居之一。

在强化学习的语境下，网络代表了一种环境——在这种环境中，状态由节点的数量和相对位置、节点之间的链路状态和数据包的动态性质决定。已被处理的数据包的终点和局部链路的状态共同形成节点的观测值。被各节点都是能选择行动的智能体。它依据策略决定应把数据包发往何处。有我们的算法计算得到的策略是随机的，与确定性相反，亦即，它会将发往相同目的地的数据包根据一定的分布经由不同的链路发送。我们实验中所考虑的策略并不决定是否接受数据包（接收控制）、应从每个邻居接受多少数据包或哪些数据包应享有优先权。

节点根据奖赏值更新其策略参数。奖赏值以在网络中分布的信号出现，该信号来自于数据包到达终点时发出的确认信息。奖赏值依赖于数据包的总传递时间。我们利用一个给定策略下的数据包的平均传递时间来衡量算法的性能（图 A.6.2 中的纵轴）。我们通过显式地惩罚路径中的绕圈现象来形成策略。我们假设每个数据包都要携带除了显然的起点和终点信息之外的有关路由历史的信息。这些信息包括数据包生成的时间、数据包最后一次被某些路由器注意到的时间、最近访问的节点的轨迹和到现在为止经历的跳数。当一个数据包被检测到在网络中花费了

过多时间而未到达终点，该数据包会被丢弃，网络也会相应地得到惩罚。因此，在我们的仿真中的决定因素是跳数是否超过了网络中的总节点数。

### A.3 算法细节

Williams 在他的强化算法中借助应用于强化学习的梯度上升介绍了策略学习的概念，而该算法被进一步总结为更广义的差错标准（由 Baird 和 Moore 提出）。整体的思路是在由经验估计的整体奖赏值的梯度方向上调整参数。我们假设这是标准的马尔科夫决策过程。我们接着考虑单智能体与部分可观测马尔可夫决策过程（POMDP）相互影响的情况。智能体的策略  $\mu$  被称为反应性策略，它由查询表组成，为每一组观测-行动（终点/链路）对赋予值  $\theta_{oa}$ 。策略将根据过往历史采取特定行动的概率定义为根据柔性最大值传输规则制定的一组参数  $\theta$  的连续可微函数，其中  $\Xi$  是温度参数：

$$\begin{aligned}\mu(a, o, \theta) &= \Pr(a(t) = a | o(t) = o, \theta) \\ &= \frac{\exp(\theta_{oa}/\Xi)}{\sum_{a'} \exp(\theta_{oa'}/\Xi)} > 0\end{aligned}\tag{A-6-1}$$

该规则确保了对于任意终点  $o$ ，任意在节点处可行的链路  $a'$  都会以与温度参数  $\Xi$  相关的小概率被选择。

我们将  $H_t$  定义为在长度  $t$  内所有可能的序列  $h = \langle o(1), a(1), r(1), \dots, o(t), a(t), r(t), o(t+1) \rangle$  的集合。为了强调某些组成元素是历史  $h$  在时间  $\tau$  的一部分，我们用  $r(\tau, h)$  和  $a(\tau, h)$  表示历史  $h$  的第  $\tau$  个奖赏和行动。我们也采用  $h^\tau$  表示序列  $h \in H_t$  在  $\tau \leq t$  的前缀： $h^\tau \triangleq \langle o(1), a(1), r(1), \dots, o(\tau), a(\tau), r(\tau), o(\tau+1) \rangle$ 。在参数  $\theta$  下遵循策略  $\mu$  所得到的值是有衰减（衰减因子  $\gamma \in [0, 1]$ ）的奖赏值的和的期望，可被写作

$$V(\theta) = \sum_{t=1}^{\infty} \gamma^t \sum_{h \in H_t} \Pr(h|\theta) r(t, h)\tag{A-6-2}$$

如果我们能计算  $V(\theta)$  对每个  $\theta_{oa}$  的导数，那么通过在步长  $\alpha$  的情况下计算更新值

$$\Delta\theta_{oa} = \alpha \frac{\partial}{\partial\theta_{oa}} V(\theta) \quad (\text{A-6-3})$$

而进行精确的梯度上升操作将是可能的。我们分析每一项权重  $\theta_{oa}$  的导数如下

$$\begin{aligned} \frac{\partial V(\theta)}{\partial\theta_{oa}} &= \sum_{t=1}^{\infty} \gamma^t \sum_{h \in H_t} \left[ r(t, h) \frac{\partial \Pr(h|\theta)}{\partial\theta_{oa}} \right] \\ &= \sum_{t=1}^{\infty} \gamma^t \sum_{h \in H_t} \Pr(h|\theta) r(t, h) \\ &\quad \times \sum_{\tau=1}^t \frac{\partial \ln \Pr(a(\tau, h)|h^{\tau-1}, \theta)}{\partial\theta_{oa}} \end{aligned} \quad (\text{A-6-4})$$

然而，考虑到强化学习本身的精神，我们假设不存在一个泛化模型使得智能体能计算  $\Pr(h|\theta)$ ，所以我们必须退回到随机梯度上升。我们采取在分布中进行采样的方法，与环境进行交互，在尝试中计算在所有时间  $t$  内的累积量

$$\Delta\theta_{oa} = \gamma^t r(t, h) \sum_{\tau=1}^t \frac{\partial \ln \mu(a, o, \theta)}{\partial\theta_{oa}} \quad (\text{A-6-5})$$

在特定的策略架构中，该方法可被轻易地转化为保证  $V(\theta)$  收敛到局部最优解  $\theta^*$  梯度上升算法。在我们所选择的策略编码方式下，我们有

$$\frac{\partial \ln \mu(a, o, \theta)}{\partial\theta_{oa}} = \begin{cases} 0, & o' \neq o \\ -\frac{1}{\bar{E}} \mu(a', o, \theta), & o' = o, a' \neq a \\ \frac{1}{\bar{E}} [1 - \mu(a, o, \theta)], & o' = o, a' = a \end{cases} \quad (\text{A-6-6})$$

将该算法应用于相互连接的控制器组成的网络，那么利用分布式梯度上升策略搜索（GAPS）的算法就基本构建完毕了。

我们将我们的分布式 GAPS 算法与其他三种方法进行了比较，如后文所述。“最佳”算法是在每条链路的代价设为一个单位时，基于最短路寻路的静态路由机制。我们将该算法也包含在内，因为它提供了最新的在工业上应用的启发式路由。“最佳容载”根据最短路算法进行路由，同时也将每个节点上的排队规模考虑进来。它接近确定性的路由算法的理论最优解，尽管实际的最优可能路由机制不再是根据网络荷载简单地要求计算最短路径，还有可能要求根据路由决策分析荷载随时间变化的情况。由于在仿真过程中每一步仿真都计算最短路在计算资源的角度上会使得代价过大，我们将进行重新调整，“最佳荷载”只有在网络中荷载量发生显著变化时才会被实现。我们将 50 个被成功传递的数据包定义为显著的荷载变化。最终，“Q-路由”是一个分布式的强化学习算法，作用于 Littman 和 Boyan 的域上。尽管我们的算法是随机的策略搜索算法，Q-路由是一种确定性的值搜索算法。注意到，我们在实现网络路由仿真时，所基于的是 Littman 和 Boyan 用于测试 Q-路由的软件。尽管如此，在“ $6 \times 6$ ”网络上，我们对“Q-路由”和“最佳”算法的仿真结果仍与 Littman 和 Boyan 有轻微的不同，可能原因在于交通建模惯例上的一些调整。例如，只有在数据包通过了整个队列到达计算资源时才认定它已被送达并可被移除，而不像最初的仿真那样，在邻居节点将数据包成功路由到终点节点是就认为数据包被送达。

我们通过一个重要的警告事件进行了对 GAPS 和上述算法的对比。GAPS 算法探索了随机策略类，而所有其他的方法都是确定性的路由策略。相应地，很自然地，我们认为 GAPS 在最优策略具有随机性的网络拓扑结构下会有更优的表现。稍后，我们会展示我们的实验证实了这项期望。

我们将分布式 GAPS 在 POMDP 的框架下进行实施。特别地，我们将每个路由器当做一个 POMDP，其中状态包括所有队列的规模、所有数据包的终点、链路的状态（上传或下载）；环境中的状态转移方程是遵循网络交通动态特性的规则；观测值  $o$  是有数据包的目的地构成；行动  $a$  是将数据包通过某一条链路传输到相邻的节点；最后，奖赏信号是单位时间内送达的数据包的平均数量。每一个智能体都利用 GAPS 强化学习算法将参数化的值沿梯度下降方向传递。已有文献表示分布式 GAPS 算法的应用将使系统整体在平稳性假设下向局部最优解收敛。该算法本质上是 Peshkin 在毕业论文第 3 章提出并在第 5 章发展的算法。

策略能通过两种不同的方式初始化：随机初始化和基于最短路初始化。我们尝试在参数空间中均匀随机选取策略进行初始化。在这样的初始化方式下，结果对学习速率非常敏感。较高的学习速率经常导致网络在组合策略空间中陷在局部

最优解，性能也不好。较低的学习速率会导致收敛过程较慢。学习速率的高与低的设定取决于网络的特性，而我们并没有找到让人满意的设定学习速率的启发式方法。显然地，类似于传递数据包的平均跳数和学习速度的特性本质上与网络节点、连接性、模性之类的特性相关。

基于这些考虑，我们采取不同的方式对控制器进行初始化。名义上而言，我们通过计算最短路由开始，将控制器设置为使多数交通量路由到最短路径，同时间歇性地发出数据包探索可行的替代链路。我们称这种方法为“ $\epsilon$ -贪婪路由”。在我们的实验中， $\epsilon$  被设置为 0.01。我们相信，该参数并不会改变结果的性质，毕竟它只在最开始的时候会影响探索行为。

算法的探索能力也以另一种方式被调节。温度和学习速率都被简单地设定为常数，一方面是为了保持系统的简单，另一方面是希望能维持控制器应对网络变化，例如链路传输失败，的能力。然而，我们的实验表明，设计机制，使得在最初的关键学习阶段之后降低学习速率能提升性能。或者，一方面在路由参数上探索不同的学习速率，另一方面探索拓扑结构的编码，将会是十分有趣的。

## A.4 实验结果

我们在几种拥有不同节点数、连接度、模数的网络上对比了路由算法。其中包括拥有 116 各节点的“LATA”电话网络。在最有网络中，GAPS 算法与其他算法表现相当或表现更优。为了阐述算法行为的关键不同与分布式 GAPS 的核心优势，我们主要研究两个仅有一条链路的位置上有所不同的网络的路由问题。

图 A.6.1 的左侧表示 Boyan 和 Littman 在实验中采用的非常规  $6 \times 6$  网络的拓扑结构。网络由两个被良好连接的部分构成，网络瓶颈在于两个桥式链路。由此导致的网络性能对荷载的依赖参见图 A.6.2 左图。所有表示性能的图形都是收敛结果，同时也是五次运行之后取的平均值。我们在荷载从 0.5 到 0.35 的情况下对网络进行测试，以与 Littman 和 Boyan 的结果进行对比。荷载服从泊松过程，而泊松过程的参数是每单位时间内平均进入网络的数据包的数量。在该网络拓扑结构下，GAPS 在低荷载情况下与其它算法相比有轻微劣势，但与最佳荷载方法在较高的荷载情况下表现相当，而比 Q-路由和最佳方法的性能更好。在低荷载情况下，性能的劣势是由于 GAPS 的探索行为导致的——一部分数据包总是被送往随机选择的链路。

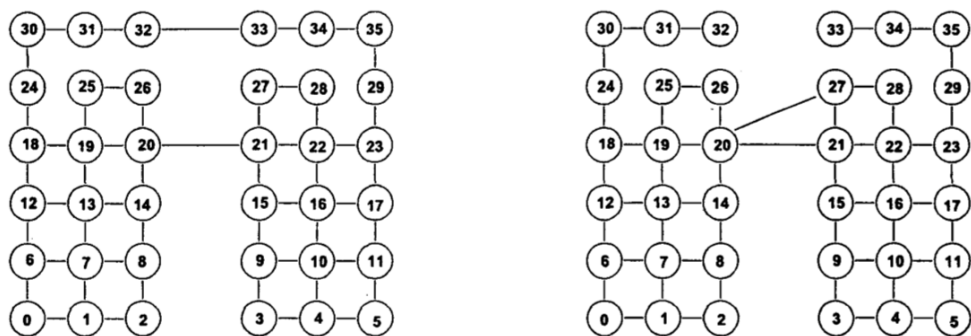


图 A.6.1 左：初始的  $6 \times 6$  网络。右：调整后的偏向于随机策略的  $6 \times 6$  网络

为了更为明确地说明算法之间的差异，我们仅通过将节点 32 与 33 之间的链路移动到节点 20 与 27 之间来调整网络，如图 A.6.1 的右侧所示。由于节点 20 显然在该结构下是瓶颈，最优路由策略必然是随机的。结果中，网络性能与荷载之间的相互关系如图 A.6.2 右图所示。GAPS 在高荷载情况下是显著优于其它算法的。它甚至在性能上超过了拥有全局信息、但被限定于确定性策略的“最佳荷载”方法。值得注意的是，确定性算法在这个网络配置中比以前的负载要低得多，因为从他们的角度来看，高度连接的组件之间的桥接比较薄（对比图 A.6.2 的左图和右图）。

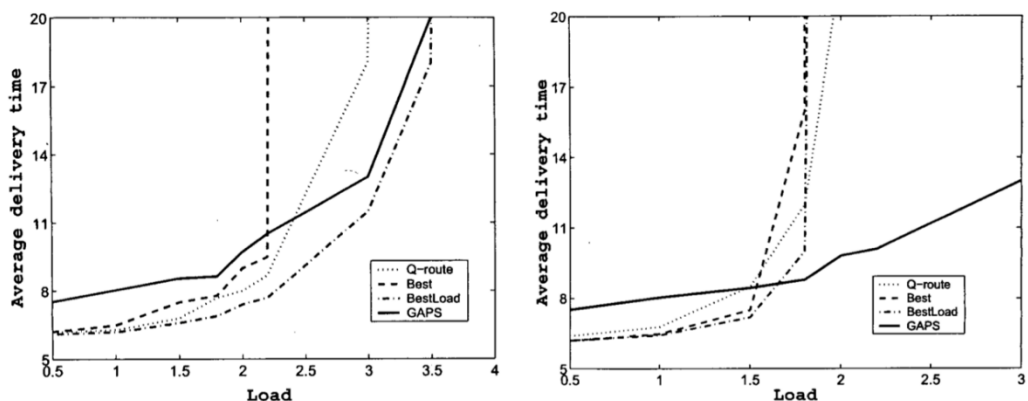


图 A.6.2 路由算法在初始的  $6 \times 6$  网络下的性能（左）和在调整后的  $6 \times 6$  网络下的性能（右）

GAPS 算法成功地适应了网络结构的变化。在荷载增加的情况下，从左边部分到右边的被偏好的路径在节点 20 的两座“桥”之间被均匀分开。在利用链路

20-27 时，相比链路 20-21，算法需要付出多经历几跳的代价，但随着节点 21 处队列长度的增加，这项代价与排队时间相比可以忽略不计。探索行为帮助 GAPS 发现链路工作状况变糟，并相应地调整策略。我们已经在每个路由器的有限状态控制器配置几个字节的内存的情况下进行了实验，但发现这并不能提升性能呢过，反而在某种程度上减缓了学习过程。

## A.5 相关工作

将机器学习技术应用到无线电通信领域是一个正在快速发展的领域。问题的主题可被归类于资源配置，例如，贷款分配、网络路由、呼叫允许控制（CAC）和电力管理。强化学习看起来能够单独地或同地时解决这些问题。

Marbach, Mihatsch 和 Tsitsiklis 已经将一种执行器-评价器（值搜索）算法应用在通信网络的资源分配问题中，方式是同时解决路由和呼叫允许控制的问题。他们采取了一种分解式的方法，将网络表示成包括拥有各自不同的奖赏的链路过程。不幸的是，在实验结果方面，即使在 4 节点和 16 节点的小规模网络上，该方法相比启发式技术，也只有很小的优势。

Carlström 介绍了基于名为预测增益调度机制的一种分解方法。呼叫允许的控制问题被分解为对未来近期的呼叫到达速率的时间序列预测和在泊松呼叫到达过程之下对呼叫策略的预计算。该方法能得到更快的学习过程，且在性能上没有损失。在一处仿真链路上，在通信承载量是 24 单位每秒的情况下，在线收敛速度提升了 50 倍。

整体而言，在通信领域，相比策略搜索算法，值搜索算法得到了更为充分的研究。值搜索(Q-学习)算法已经取得了非常有希望的结果。Boyan 和 Littman 的算法——Q-路由方法，已经证明了比基于最短路非自适应性技术更为优越，同时在网络动态变化的情境下健壮性也更好，场景包括一个非常规的  $6 \times 6$  网格和拥有 116 个节点的 LATA 电话网络。它对数据包所要经过的节点的数目与产生拥堵的可能性之间的权衡进行调节。

Wolpert, Tumer 和 Frank 构建了一种用于互联网交通路由的被称为群体智慧(COIN)的神经网络。该方法包括根据全局效用信息和观测到的局部动态信息自动地初始化并更新强化学习智能体(节点)的局部效用函数。他们的仿真结果在由七个节点组成的样例网络中优于完全知识最短路算法。COIN 网络采用了一



种在精神上与本项研究较为相似的方法。他们依赖于一种在无需给每个智能体节点明确的网络拓扑结构信息下就能收敛到局部最优解的分布式强化学习算法。然而，COIN 与我们的方法的不同之处在于，其需要将初始网络结构的信息引入，方式是将网络分成半自发的共享局部效用函数并鼓励合作的邻居。相反，我们网络中的所有节点则直接通过全局奖赏值更新它们的算法。

本文所展示的工作集中在利用策略搜索的方法进行数据包路由。该工作与 Tao, Baxter 和 Weaver 的工作较为类似。他们应用了策略梯度算法，以诱使分组交换网络中的节点相互合作以最小化平均数据包延时。尽管他们的算法在几种网络类型中表现良好，仍需要耗费许多（上万）次的尝试才能在一个仅有几个节点的网络中达到收敛。

将强化学习应用到通信经常包括在多种评价准则下优化性能。最近的一项关于这个富有挑战性的问题的讨论可以参见 Shelton 的工作。在无线通信方面，该问题由 Brown 得到解决。他考虑的问题是寻找能够同时地最大化奖励值的能量管理策略，该奖励值奖励的是利用尽可能少的电池用量提供通信。问题被定义成折扣无界的随机最短路，其中折扣因子会随模型中电量的损失而变化。该方法最终在电量使用方面收获了显著（50%）的提升。

Gelenbe 等也将奖赏当做数据报丢失和延迟的概率加权组合进行计算。在认知数据包网络中，数据包自身也是能进行路由和流量控制的智能体。他们将数据包分成三类：“聪明”、“愚笨”和“确认”。一小部分的聪明数据包会学习最有效的遍历网络的方法，愚笨数据包则是简单地跟从聪明数据包所采取的路径，而确认数据包则采用与聪明数据包相反的路径向愚笨数据包传递路由信息。聪明数据包与愚笨数据包之间的分歧是探索/利用困境的明确表现。聪明数据包允许网络对结构变化做出适应调整，而愚笨数据包则利用了变化之间的相对稳定性。该工作在由 100 各节点组成的仿真网络和由 6 台电脑组成的物理网络都取得了理想的结果。

Subramanian, Druschel 和 Chen 采用了与蚁群在思想上相近的方法。网络中主机个体会保留与其他主机通信时的代价路由表（例如其应该遍历哪些路由器，而代价有多昂贵）。这些表格会周期性地被“蚂蚁”——用于评价遍历主机间链路的代价的信息——更新。蚂蚁们以一定概率地在可行路径间被导向。蚂蚁们会将相关传输代价通知沿途的主机。主机们利用这些信息，根据更新规则调整自己的路由表。有两种类型的蚂蚁，普通的蚂蚁利用主机的路由表来调整自己被定向到特定路径上的概率。在经过多次尝试后，所有拥有相同任务的普通蚂蚁开始采

用相同的路径。它们的功能在于主机表中的代价值在网络稳定的情况下收敛到正确的数字。同质蚂蚁则以相等的概率选择任何路径。这些蚂蚁会继续探索网络以保证能对链路状态或链路代价的变化作出成功的适应。

## A.6 讨论

应该承认的是，本工作对网络路由过程的仿真离实用的距离还很远。一个更为实际的模型可以考虑类似于在链路贷款和路由节点缓存大小限制方面的非同质网络、数据包的冲突、数据包命令限制、商用与政府子网络相关的链路的代价、最小服务质量要求等。个体数据包优先级的引入也带来了另一组优化问题。然而，我们应用的学习算法体现出了在解决适应性电信通讯协议方面的希望，同时存在着几种显然的拓展本研究的方向。将域的知识整合进控制器结构就是该类型的方向之一。该方法会包含以分层的方式将节点分类成子网络和路由数据包。在此方向上更进一步，我们可以尝试将学习算法部署到自组网的路由中。自组网是节点能从系统中被动态地引入或终止的网络，同时，仍活跃的节点会移动、断开某些连接再建立新的连接。一个现实假设是，网络的物理变化速度慢于路由与演化。在该假设下，自适应路由协议一定会比任何启发式预定义路线表现更好。我们目前正在该方向的研究。

## 原文索引

- [1] Peshkin L, Savova V. Reinforcement learning for adaptive routing[C]. International Joint Conference on Neural Networks. IEEE, 2007:1825-1830.