

基于顺序选择的自动驾驶车辆十字路口调度方案

吕 品^{1,2}, 何岳滨¹, 许 嘉^{1,2}

(1. 广西大学计算机与电子信息学院, 广西南宁 530004; 2. 广西多媒体通信与网络技术重点实验室, 广西南宁 530004)

摘 要: 自动驾驶车辆的普及是城市交通发展的重要趋势, 然而以交通灯为代表的现有调度方案难以指导这些车辆高效通过十字路口. 为此, 本文设计了一种基于顺序选择的无交通灯十字路口调度方案以提高自动驾驶车辆通行效率. 首先根据车辆的物理性能对其到达路口的最早时间点进行估算, 然后在此基础上寻找车辆可行的计划到达时间点, 最终根据计划到达时间为每辆车制定到达路口的行驶计划. 在 SUMO (Simulation of Urban MObility) 平台上进行的大量仿真实验验证了本文提出方案的有效性. 从实验结果可以看出, 本文所提出的方案使自动驾驶车辆在调度区平均行驶时间等指标上均优于对比方案; 特别是在路口面临高压车流时, 本文所提出方案的优势更加明显.

关键词: 自动驾驶车辆; 顺序选择; 调度; 交通效率

中图分类号: TP273 **文献标识码:** A **文章编号:** 0372-2112 (2021)05-0912-08

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.12263/DZXB.20200956

Sequential Selection-Based Scheduling for Connected and Automated Vehicles at Intersections

LÜ Pin^{1,2}, HE Yue-bin¹, XU Jia^{1,2}

(1. School of Computer and Electronics Information, Guangxi University, Nanning, Guangxi 530004, China;

2. Guangxi Key Laboratory of Multimedia Communications and Network Technology, Nanning, Guangxi 530004, China)

Abstract: Connected and automated vehicles (CAVs) will become the mainstream of urban traffic. However, the existing scheduling schemes, such as traffic lights, are difficult to guide CAVs to pass through intersections efficiently. In order to improve vehicle traffic efficiency, a scheduling scheme based on sequential selection is designed for intersections without traffic lights. A feasible time for a vehicle to arrive at the intersection is planned according to its physical abilities and status of other CAVs. Extensive simulation experiments are conducted on the SUMO platform to verify the effectiveness of the proposed scheme. From the experimental results, it is revealed that the proposed scheme improves the traffic efficiency at intersections, comparing with other methods. Especially, when the traffic load is heavy, the performance gain of the proposed scheme is more obvious.

Key words: connected and automated vehicles (CAV); sequential selection; scheduling; traffic efficiency

1 引言

近年来, 自动驾驶技术成为智能科学与道路交通领域的研究热点. 与传统车辆相比, 自动驾驶车辆通常在通信能力和计算能力方面具有显著的优势, 使得自动驾驶车辆能够更加安全、高效地行驶^[1,2]. 因而在一些工业园区或旅游景点, 已经出现了完全由自动驾驶车辆组成的交通应用场景. 可以预见, 自动驾驶车辆将成为未来城市交通的主体.

十字路口往往是城市交通中的关键节点. 长期以来, 对于车辆在十字路口的调度主要依赖交通灯. 然而, 交通灯的状态转换时间通常是固定的, 难以根据路况进行动态调整, 导致路口通行效率较低, 严重时会造成交通拥堵. 不仅如此, 在自动驾驶车辆进一步普及的未来, 这类调度方案难以发挥车辆的通信与计算优势.

为了解决上述问题, 本文提出了一种基于顺序选择的调度机制. 该机制以“先进先选” (First In First Select, FIFS) 为策略, 分别为每辆即将到达十字路口的车

辆规划符合其自身物理性能的路口通过方案. 方案充分考虑了之前到达的车辆如何通过路口, 在避免交通事故的前提下, 尽量提升路口通行效率. 实验表明这种调度机制可以在车辆到达率较大时达到更高的十字路口通行效率.

2 相关工作

2.1 基于交通灯的调度方法

早期的交通灯调度方法^[3]根据交通需求人工确定相位持续时间, 这种方法很难灵活应对各种不同的道路状况. 随着计算机性能不断提升, 基于控制论的方法^[4]与使用 DQN (Deep-Q-learning) 深度强化学习等人工智能技术的方法相继被提出. 其中, 文献^[5]所提出的 DQN 相关方法在提高车辆吞吐量、减少延迟方面取得了较好结果. 文献^[6]改进了这一思路, 增加交通灯当前相位作为路况特征, 降低了车辆通过十字路口的平均时间. 文献^[7]方法则是采用模糊逻辑, 进入十字路口的各路车流依据其等待队列长度被划分到五个等级的模糊集中, 并根据所划分的模糊集进行相位决策. 上述方法主要依赖交通灯作为与车辆通信的方式, 使得这类方法难以进行精确调度, 一辆汽车的行动延迟有可能导致路口拥挤甚至阻塞. 与此同时, 这类调度方法缺乏根据路况对附近车辆进行安全行驶指导的手段.

2.2 协同调度方法

为了解决基于交通灯的调度方法所带来的各种问题, 利用车联网通信进行车辆调度成为了新的研究热点. Dresner 等人提出了一种时间预留方案^[8]: 车辆不断向协调器发送请求信号直至获得在某一时间段进入路口的许可, 并在获得许可后宣告占有该时间段. 文献^[9]提出可以制定一种排序规则, 使协调器可以凭借该规则统一处理并直接指定车辆的到达时间段. 之后, Perronet 等人则进一步发展并实现了协调器集中处理的方案, 提出了一种相对完整的调度服务协议^[10]. 研究工作文献^[11~13]均提出了分布式控制方案, 并致力于在控制区内部提出一种针对某单一指标最优的控制方案.

在各种无交通灯调度算法提出的假设中, “先进先出”(First In First Out, FIFO) 被广泛使用. 这种假设使得无交通灯算法的相关研究得以在一个较为简单的交通环境下进行. 但是正如文献^[14]中所提, FIFO 过于保守以至于限制了路口吞吐量的进一步提升. 其他的针对 FIFO 的解决方案还有混合线性规划方法^[15]、优先级方法^[16]等. 但是, 这些方案由于采用整体指标而忽略了每辆车自身的运动情况, 有时可能产生消极的决策.

3 系统模型

考虑如图 1(a) 中所示的一个十字路口, 其中心为

O . 两条相互垂直的、中心线过 O 的双向单车道道路交叉于汇合区域(由方形虚线标出, 是边长为 S 的正方形). 一个协调器通过与到达的车辆通信进行车辆调度, 其作用范围由圆形虚线标出. 调度区域的各个入口与汇合区域的距离相同, 记为 L .

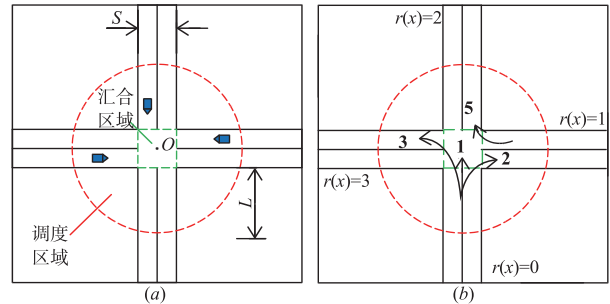


图1 十字路口与自动驾驶车辆模型

车辆 x 进入调度区域时明确自己唯一的行驶路线 $z(x) = G(x) + 3r(x)$, 其中 $r(x) \in \{0, 1, 2, 3\}$ 是车辆通过路口前所在道路的编号, 分布如图 1(b) 所示; $G(x) = \{1, 2, 3\}$ 分别代表车辆的目标方向(直行, 右转或者左转). 例如, 假设车辆 x 的出发道路为 $r(x) = 1$, 目标方向是右转则 $G(x) = 2$, 则其行驶路线编号 $z(x) = 2 + 3 \times 1 = 5$. 类似地, 可以计算得知 $z(x) \in \{1, 4, 7, 10\}$ 时车辆为直线行驶, 其他路线则包含转向过程.

车辆与协调器进行信息交换及处理的具体步骤如下.

步骤 1 车辆将自身的序号与预期行驶方向上传至协调器, 协调器则将当前时间 t 与调度区域内所有车辆的信息总集(以下简称为路况总集) $B(t)$ 传输给车辆, 其中每一辆车的信息集表示为 $b(x) = \{x, T(x), z(x)\} \in B(t)$, $T(x)$ 是车辆预计到达路口的时间;

步骤 2 车辆计算出到达路口的预定时间, 并制定运动计划, 最终生成属于自身的信息集并上传给协调器;

步骤 3 协调器将此车辆的信息集加入到 $B(t)$ 中, 并根据所有车辆到达时间的先后顺序重排 $B(t)$, 车辆则开始执行行驶计划. 当车辆离开调度区域后, 协调器将从总集中删除该车辆相关的信息集.

本文的重点是解决上述过程中的步骤 2. 在这一计算过程中, 避免与其他车辆产生冲突是需要考虑的关键因素之一. 将车辆之间可能产生的冲突进行分类(将在第 4 节进行详细说明), 记为 $c(x, y) \in \{0, 1, 2\}$, 其中 x 与 y 代表两辆不同的车. 另一个因素是得到初始方案所需要的物理约束, 在汇合区域外, 车辆速度需满足式(1)所示的约束:

$$0 < v(x, t) < v_{\max}(x) \quad (1)$$

其中 $v(x, t)$ 是车辆 x 在时间 t 的速度(汇合区域外),

$v_{\max}(x)$ 是由车辆自身与道路决定的极限速度.在汇合区域内,车辆速度约束由式(2)表达:

$$0 < v_{\text{cross}}(x, t) < v_c(x) \quad (2)$$

$v_{\text{cross}}(x, t)$ 是车辆 x 在汇合区域内的速度, $v_c(x)$ 是车辆在汇合区域内的速度上限.如果车辆是直行经过路口,其速度上限即为车辆能够达到的最大速度 $v_{\max}(x)$;如果车辆经过路口时需要转向,则其速度上限是 $\sqrt{\mu g R}$,转弯时超过此速度则容易发生事故. $v_c(x)$ 由式(3)定义:

$$v_c(x) = \begin{cases} v_{\max}(x), & z(x) \in \{1, 4, 7, 10\} \\ \sqrt{\mu g R}, & \text{else} \end{cases} \quad (3)$$

其中, μ 是地面摩擦系数, g 为重力加速度, $R \in \{R_1, R_2\}$ 为转弯半径. R_1 与 R_2 分别是右转弯道(例如图 1(b)中方向 2 所在路线)与左转弯道(例如图 1(b)中方向 3 所在路线)的半径.

4 基于顺序选择的车辆调度方案

本节将介绍一种基于顺序选择,即 FIFS(First In First Select)的十字路口调度方案.在此之前,需要提出一些必要的定义.

定义 1 在图 1(a)所定义的模型下,路线 z_1, z_2 之间的关系 $e(z_1, z_2)$ 属于且仅属于以下的三个子集之一.

- (1) X : 选择 z_1 与 z_2 的车辆存在侧撞的可能.
- (2) Λ : 选择 z_1 与 z_2 的车辆存在追尾的可能.
- (3) K : z_1 与 z_2 没有相交之处,分别以它们作为规划路径的车辆之间没有碰撞风险.

在此基础上,可以确定任意两辆车 x 与 y 的冲突类型,如式(4)所示:

$$c(x, y) = \begin{cases} 0, & e(z(x), z(y)) \in X \\ 1, & e(z(x), z(y)) \in \Lambda \\ 2, & e(z(x), z(y)) \in K \end{cases} \quad (4)$$

定义 2 不同的两辆车 x, y 之间在相互之间不存在碰撞风险的前提下,计划到达时间的最小间隔称为 $d(x, y)$,其值由两车之间的冲突类型决定.具体定义如式(5)所示:

$$d(x, y) = \begin{cases} u_0/v_x, & c(x, y) = 0 \\ \delta(x, y)/v_y, & c(x, y) = 1 \\ 0, & c(x, y) = 2 \end{cases} \quad (5)$$

其中 $\delta(x, y)$ 为两车避免追尾的安全距离, v_x 与 v_y 分别是车辆 x 与 y 到达路口时的预计速度, u_0 为 x 的路线在汇合区域内部分的长度.

以下将根据车辆进行处理的顺序介绍整个方案,4.1节介绍车辆如何计算在理想情形下最早到达路口的时间;4.2节提出在已知理想到达时间与路况的前提下对车辆进行到达时间调度的算法.

4.1 理论到达时间计算

对车辆 x 来说,在不考虑其他车辆的假设下自身能够到达路口的最早时间称为理论到达时间 $T_0(x)$.对 $T_0(x)$ 的计算根据车辆的路线类型不同(直行/弯道)分为两种情形,将在本节分别介绍.

4.1.1 直行车辆理论到达时间

(1) x 能够在到达路口之前加速到极限速度,该条件表达为 $(v_{\max}^2(x) - v_0^2(x))/(2a_x) \leq L$,其中 a_x 为车辆 x 的加速度, $v_0(x)$ 是车辆进入调度区域时刻的速度.在这种情形下,车辆将持续加速直至到达极限速度,随后保持速度巡航至路口汇合区域,如式(6)所示:

$$T_0(x) = \frac{v_{\max}(x) - v_0(x)}{a_x} + \frac{L}{v_{\max}(x)} - \frac{v_{\max}^2(x) - v_0^2(x)}{2a_x v_{\max}(x)} \quad (6)$$

(2) x 无法在到达路口之前加速到极限速度,即服从条件 $(v_{\max}^2(x) - v_0^2(x))/(2a_x) \geq L$.此时车辆全程进行加速,如式(7)所示:

$$T_0(x) = \sqrt{v_0^2(x) + 2a_x L} - v_0(x)/a_x \quad (7)$$

4.1.2 转弯车辆理论到达时间

情况 1 车辆能够在行驶过程中加速到极限速度,并在到达汇合区域前安全地减速至转向速度 $\sqrt{\mu g R}$ (在第 3 节中说明).此时车辆的初速度、最大转向速度、最大直行速度应满足约束条件: $(2v_{\max}^2(x) - v_0^2(x) - v_c^2(x))/(2a_x) < L$,此时车辆的理论到达时间表达为式(8)所示:

$$T_0(x) = \frac{2v_{\max}(x) - v_0(x) - \sqrt{\mu g R}}{a_x} + \left(L - \frac{2v_{\max}^2(x) - v_0^2(x) - \mu g R}{2a_x} \right) / v_{\max}(x) \quad (8)$$

情况 2 在情况 1 中的条件无法满足时,则以式(9)估计车辆理论到达时间:

$$T_0(x) = \frac{\sqrt{4a_x L + 2v_0^2(x) + \mu g R} - v_0 - \sqrt{\mu g R}}{a_x} \quad (9)$$

通过上述计算获得车辆的理论最早到达时间点之后,车辆 x 即可据此生成临时信息集,即 $b_m(x) = \{x, T_0(x), z(x)\}$,然后将其加入来自协调器的本时刻路况数据总集 $B(t)$ 中,并将路况总集根据到达时间按先后顺序进行重排序,形成含有车辆 x 信息的有序临时集合,记为 $B_m(x, t)$.

4.2 到达时间调度算法

4.2.1 算法原理

算法整体思想如下:对待调度的车辆 x ,首先以其理论最早到达时间 $T_0(x)$ 作为备选到达时间,与此同时考虑根据 4.1 节生成的临时集合 $B_m(x, t)$,并在符合此有序临时集合所描述的车辆的到达顺序下,尝试为 x 获取取

可能早的计划到达时间 $T(x)$. 如果在当前的顺序下车辆 x 面临无法避免的碰撞风险,则算法将向后生成新的备选时间,更新车辆 x 的临时信息集 $b_m(x)$ 并重排临时集合 $B_m(x,t)$. 在此之后进行新一轮的尝试,如此循环直至确定车辆的计划到达时间. 图 2 展示了算法的执行流程.

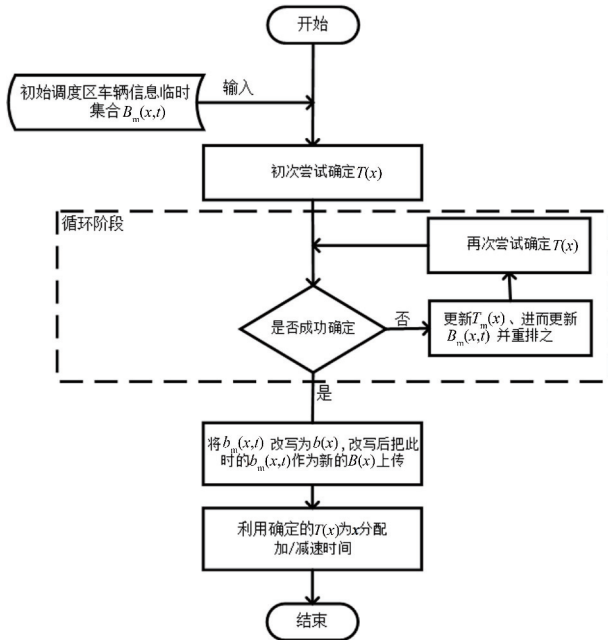


图2 到达时间调度算法流程

4.2.2 首轮执行

首轮执行中, 备选时间 $T_m(x) = T_0(x)$ (以下直接以 $T_0(x)$ 表示). 在此基础上, 展开算法首轮执行中对计划到达时间的讨论.

情形 1 若车辆 x 不存在前置车辆与后置车辆(前置车辆指在当前 $B_m(x,t)$ 中位序在 x 之前、并且与 x 道路不同的车辆; 后置车辆指在当前 $B_m(x,t)$ 中位序在 x 之后的车辆). 此时 x 为调度区域内的唯一车辆, 直接将当前备选到达时间作为车辆 x 的计划到达时间, 即 $T(x) = T_0(x)$.

情形 2 若车辆 x 仅存在前置车辆, 此时车辆在确定前置车辆 p_x 之后, 将通过定义 2 获得自身与前置车辆的最小安全时间间隔 $d(p_x, x)$. 此后在保证两车保持安全时间间隔的基础上, 车辆将在 $[T(p_x), +\infty]$ 中选择尽可能早的时间点作为计划到达时间, 可以表达为式(10):

$$T(x) = \text{MAX}(T(p_x) + d(x, p_x), T_0(x)) \quad (10)$$

情形 3 若车辆 x 仅存在后置车辆, 则 x 在确定后置车辆 n_x 之后首先考虑与后置车辆的冲突关系与当前两者的时间间隔(记为 $l(x, n_x)$):

(i) 若当前时间间隔 $l(x, n_x)$ 服从条件 $l(x, n_x) > d(x, n_x)$, 此时车辆 x 与 n_x 不存在碰撞风险, 故可将备选时间点直接作为计划到达时间点.

(ii) 若当前时间间隔 $l(x, n_x)$ 服从条件 $l(x, n_x) \leq d(x, n_x)$, 保持当前顺序将与已经确定计划到达时间的 n_x 发生碰撞, 更新 $T_m(x)$ 并进入下一轮算法执行.

情形 4 若车辆 x 同时存在前置车辆与后置车辆, 车辆 x 首先获取 p_x 与 n_x , 然后根据两者的计划到达时间间隔 $l(p_x, n_x)$ 进行进一步判断:

(i) $l(p_x, n_x) < d(x, p_x) + d(x, n_x)$, 此时在保留当前的 $B_m(x,t)$ 中元素顺序的前提下无法确定计划到达时间, 需要更新备选时间, 循环执行算法进行判断.

(ii) $l(p_x, n_x) > d(x, p_x) + d(x, n_x)$, 并且此时 $T_0(x)$ 的值服从式(11):

$$T_0(x) \in [T(n_x) - d(x, n_x), T(n_x)] \quad (11)$$

即 $T_0(x)$ 使 x 与后置车辆存在碰撞风险, 类似于情形 3-(ii), 无法在当前顺序下确定计划到达时间.

(iii) $l(p_x, n_x) > d(x, p_x) + d(x, n_x)$, 并且此时 $T_0(x)$ 的值服从式(12):

$$T_0(x) \in [0, T(p_x) + d(x, n_x)] \quad (12)$$

此时类似于情形 2, 车辆将选择一个尽可能早且安全的时刻作为计划到达时间.

4.2.3 迭代执行

对于在首轮执行中无法确定车辆计划到达时间的情形, 需要进入图 2 所示的循环阶段以进一步尝试. 以下将分别介绍循环阶段的两个关键步骤: 更新与重新判断.

4.2.3.1 更新

更新分为以下三个步骤:

(1) 将车辆的备选到达时间 $T_m(x)$ 移动至当前后置车辆的到达时间之后, 并且恰好与后置车辆保持最小安全间隔.

(2) 改变当前备选时间之后, 由于实际考虑的车辆到达顺序已经变化而需要进行的车辆临时信息总集 $B_m(x,t)$ 的更新.

(3) 对于更新后的 $B_m(x,t)$, 车辆 x 重新识别前置车辆与后置车辆.

4.2.3.2 重新判断

由于在更新过程中已经使得当前车辆 x 的备选到达时间与前置车辆(原后置车辆)保持最小安全时间间隔, 故对于车辆 x 来说只需要考虑新后置车辆的情形:

(1) 后置车辆不存在, 可归为 4.2.2 节中情形 2;

(2) 后置车辆存在且不等式(13)成立:

$$T_m(x) > T(n'_x) \quad (13)$$

其中 $T(n'_x)$ 为更新后车辆的后置车辆, 此时为 4.2.2 中情形(4)-(i)的一种实例, 无法本轮完成计划到达时间的确定, 进入下一轮循环;

(3) 后置车辆存在且不等式(13)不成立, 此时为 4.2.2 节中情形 4-(iii)的一种实例, 将依据相同的处理

方法确定车辆 x 的计划到达时间.

4.2.4 后续处理与车辆行驶规划

通过 4.2.2 节或 4.2.3 节所描述的方法获得计划到达时间 $T(x)$ 之后,车辆将根据 $T(x)$ 为自身制定完整的行驶计划,具体如下所示.

(1) 如果车辆沿直线前进,车辆明确自身加/减速应持续的时间 T_{acc} ,如式(14)所示:

$$T_{acc} = \frac{\sqrt{a_x^2 t_1(x)^2 + 2v_0(x)a_x t_1(x) - 2a_x L + a_x t_1(x)}}{a_x} \quad (14)$$

其中 $t_1(x) = T(x) - t_0(x)$ 是 x 计划从进入调度区到路口经历的时间.在加速 T_{acc} 之后, x 将保持速度直至离开汇合区域.

(2) 车辆需要转向变道,则其经历一个加/减速-匀速-加/减速的过程.第一段变速持续时间如式(15)所示:

$$T_{acc,i} = (v_c(x) - a_x t_1(x) + \sigma(x)) \quad (15)$$

其中 $\sigma(x) =$

$$\sqrt{(v_0(x) + v_c(x) + a_x t_1(x))^2 - 4a_x L + 2(v_0^2(x) + v_c^2(x))},$$

而匀速阶段持续时间为

$$T_{route} = (a_x + 1) t_1 \pm \omega(x) - \frac{v_0(x) + v_c(x)}{2} \quad (16)$$

其中 $\omega(x) =$

$$\sqrt{(v_0(x) + v_c(x) + a_x t_1(x))^2 - 4a_x L - 2(v_0^2(x) + v_c^2(x))}.$$

若第一段变速为加速则上式 $\omega(x)$ 项取负,反之取正.在经历匀速过程之后,车辆将匀减速直至汇合区域.

5 实验

5.1 实验参数

本实验在 SUMO(Simulation of Urban Mobility) 平台上进行,该平台为各种车辆行为控制、运动控制提供了大量易于使用的 API.首先在该平台上搭建如图 1(a) 所示的路网结构,初始参数如表 1 所示.

其中,道路限速指车辆在各车道能达到的最高速度,车辆安全距离指同一车道上两辆车为了避免追尾至少应保持的最短距离.

表 1 模型参数

模型参数	默认取值
道路限速	30m/s
右转弯半径 R_1	6m
左转弯半径 R_2	9m
车辆安全距离(防止追尾)	10m
调度区入口至汇合区域距离 L	275m
汇合区域边长 S	15m

5.2 实验参数

使用以下指标来评估调度方案的性能.

车辆通过率 在测试时间内所有通过路口的车辆数量与出发车辆数量的比值,该值越高,说明交叉路口的整体通行效率越高.

调度区平均行驶时间 车辆从进入调度区域开始到进入汇合区域截止,所用的时间称为调度区行驶时间.这项指标是所有实验车辆调度区行驶时间的平均值,该值越低代表车辆能够更快地通过十字路口.

平均延迟比率 D 反映进入调度区的车辆对自身的速度性能利用率指标,由式(17)定义:

$$D = \frac{1}{n} \sum_{i=1}^n (1 - \bar{v}(x_i)/v_{max}(x_i)) \quad (17)$$

其中, $\bar{v}(x_i)$ 表示车辆 x_i 在调度区内的平均速度, n 是车辆总数.平均延迟比率越低,代表调度区内车辆能够发挥更接近自身上限的速度性能.

5.3 对比方法

为了评估模型的有效性,依据 5.2 节所提指标将本文所提出的模型与下列两种基准方法进行比较,并尽可能调整各方法参数,使它们拥有更好的表现.

RA(reservation algorithm) 以文献[8]所描述方法为例,这种调度方法采用车辆进入调度区域时的速度估计到达时间,并以此比对各车辆之间的冲突关系进行调解并确定每一辆车恒定的移动速度.最后令车辆匀速巡航到达路口.与 RA 方法对比是为了验证本文提出的时间调度算法的性能.

FIFO-RA 调度方法 以文献[10]所描述方法为例,这类方法中车辆根据自身进入调度区域的顺序唯一确定自身在 FIFO 队列中的位置,并仅凭借 FIFO 队列中前一辆车的预计到达时间确定自身的到达时间,以此作为参考进行速度调整与行驶.与 FIFO-RA 方法对比是为了验证本文提出的 FIFS 策略的有效性.

5.4 实验设置

在本实验中,沿该道路到达十字路口汇合区域的车辆总数服从泊松分布.除此之外,为了模拟在实际道路中出现的拥有不同加速性能与最大速度的车辆,实验设置五种性能不同类型的车辆,如表 2 所示.

表 2 车辆类型

类型编号	最大速度(m/s)	最大加速度(m/s ²)
1	20.0	1.00
2	22.5	1.25
3	25.0	1.50
4	27.5	1.75
5	30.0	2.00

在此基础上,分为以下两组数据进行实验:

实验组 1 在所有车道的出发概率相同的前提下,改变车辆到达率,以此观察模型与对比方法的各项指标在均衡车流的情况下,随车辆到达率而变化的情况。

实验组 2 在车辆总到达率不变(= 0.72)的前提下,改变南北方向与东西方向车道出发率的差值以观察模型与对比方法的各项指标在相同车辆总到达率的条件下,随车流不平衡程度的变化情况。

5.5 实验结果与分析

5.5.1 实验组 1

从图 3 和图 4 可以看出,FIFS 方法在调度区平均行驶时间与平均延迟上相较于 FIFO-RA 与 RA 方法均有明显优势.例如车辆到达率为 1.14 车/s 时,FIFS 的这两项指标分别比 FIFO-RA 降低 6.90% 与 21.55%,比 RA 方法则分别降低 16.02% 与 41.95%.从图 5 中可以看出,FIFO-RA 方法与 RA 方法在车辆通过率上较为接近,而 FIFS 方法的通过率更高。

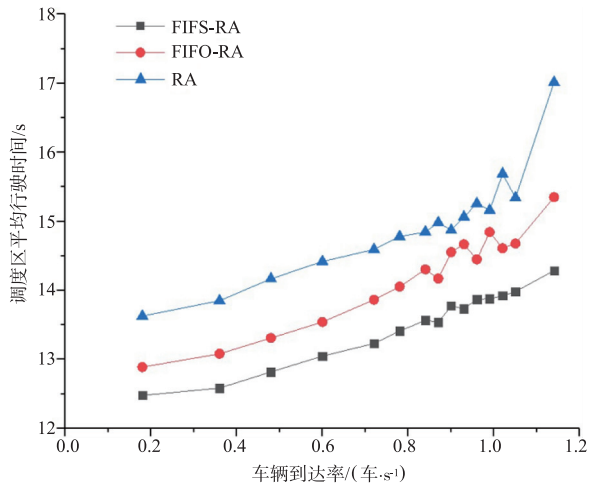


图3 调度区平均行驶时间随车辆到达率的变化情形

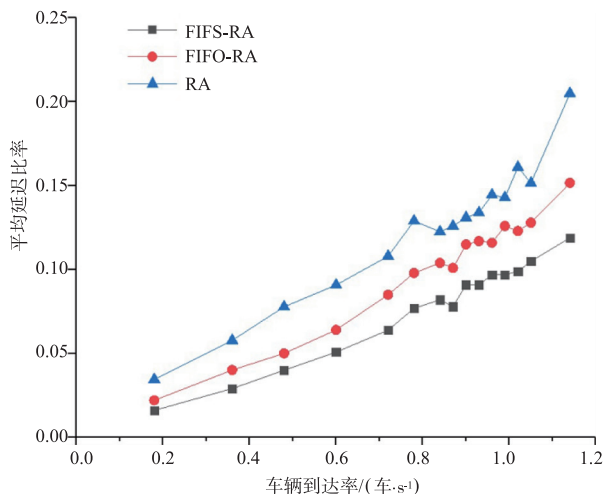


图4 平均延迟比率随车辆到达率的变化情形

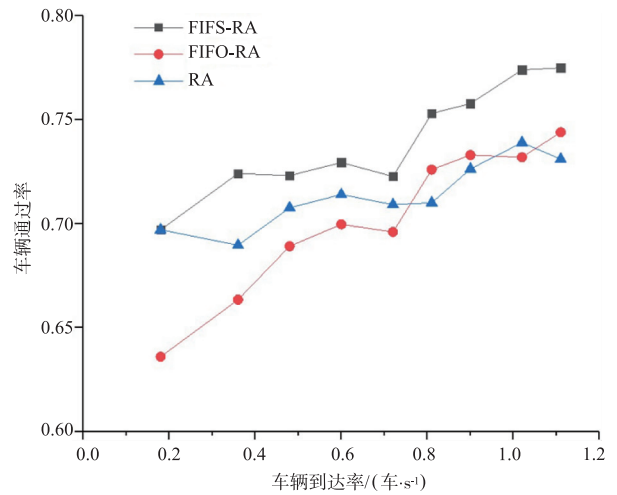


图5 车辆通过率随车辆到达率的变化情形

值得注意的是,随着车辆到达率的增大,各方法受到车辆到达率的影响程度不同,例如在车辆到达率提升到一定程度情形下,对比方法在调度区延迟与平均调度区行驶时间上均有较大幅度的上升,FIFS 方法仍然较为平稳.而在车辆通过率方面,随着车辆到达率的增加各种方法的调度效果开始显现,使得路口资源能够被更充分地利用,从而使整体车辆通过率得以提升.通过三种方法的对比可知,本文所提出 FIFS 方法凭借更加高效的调度策略取得了更高的通过率。

5.5.2 实验组 2

从图 6 和图 7 可以看出,随着车辆到达率偏移的上升,FIFO-RA 与 FIFS 方法在调度区平均延迟上有一定下降.主要原因是车流集中于某一个方向的程度越高,车辆之间冲突的类型就越倾向于追尾风险与无风险,总体上车辆之间的最小安全间隔降低(见定义 2),从而使车辆能够更高效地通过路口,RA 方法由于类似的原因平均延迟有所下降,但调度区行驶时间的增高是因

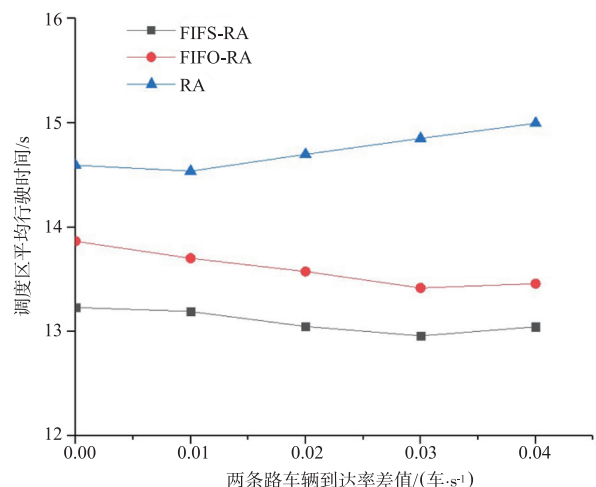


图6 调度区平均行驶时间随车流不平衡程度的变化情形

为随着车流压力的集中,在车辆直线行驶的过程中产生了更多的停顿,而不仅仅是在汇合区域中。

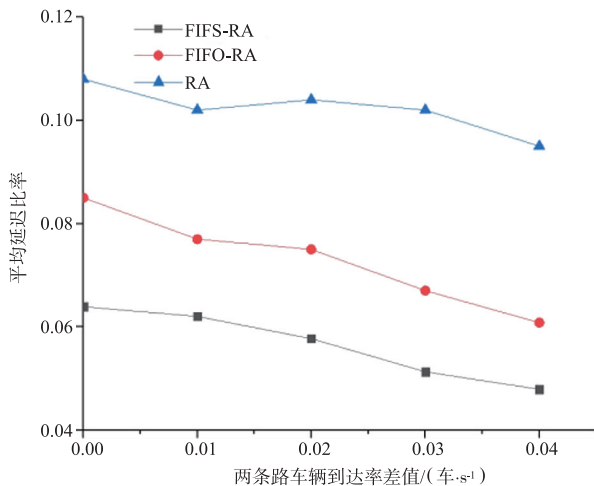


图7 平均延迟比率随车流不平衡程度的变化情形

图8则分别展现了各方法的车辆通过率随车辆到达率不平衡程度的影响。可以看到,在车辆到达率偏移较低时三种方法效果相近,而在车辆到达率偏移增加时,FIFS方法展现出明显优势。

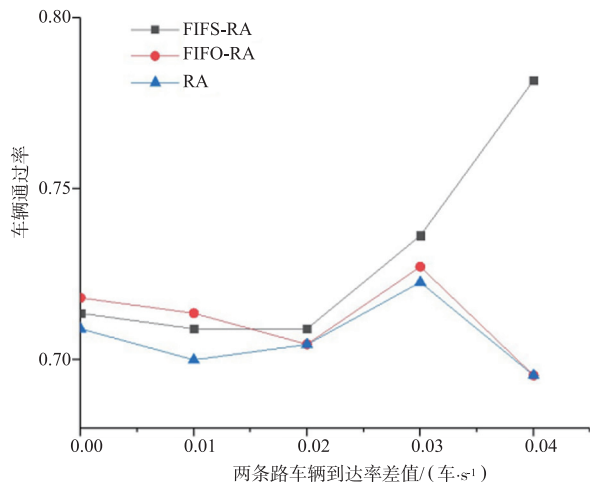


图8 车辆通过率随车流不平衡程度的变化情形

6 总结

本文提出了一种基于顺序选择的自动驾驶汽车在十字路口的调度方案。首先提出的是车辆与协调器的信息交互方式与数据存储结构;然后设计算法使车辆能够在当前路况中选择能够最快到达的时间点,并依此进行运动规划;最后在仿真平台上设计实验证明了本文所提出方案的有效性。这项研究仍存在一些局限性:例如未考虑车辆的优先级与相关的判定机制,未考虑对不平衡车流进行针对性的调整等等,这些方向将在之后的工作中进行研究与探索。

参考文献

- [1] Héry E, Xu P, Bonnifait P. Along-track localization for cooperative autonomous vehicles [A]. 2017 IEEE Intelligent Vehicles Symposium (IV) [C]. Los Angeles, USA: IEEE, 2017. 511 – 516.
- [2] 黄志清, 曲志伟, 张吉, 等. 基于深度强化学习的端到端无人驾驶决策 [J]. 电子学报, 2020, 48(9): 1711 – 1719.
Huang Z Q, Qu Z W, Zhang J, et al. End-to-end autonomous driving decision based on deep reinforcement learning [J]. Acta Electronica Sinica, 2020, 48(9): 1711 – 1719. (in Chinese)
- [3] Miller A J. Settings for fixed-cycle traffic signals [J]. Journal of the Operational Research Society, 1963, 14(4): 373 – 386.
- [4] Fleck J L, Cassandras C G, Geng Y F. Adaptive quasi-dynamic traffic light control [J]. IEEE Transactions on Control Systems Technology, 2016, 24(3): 830 – 842.
- [5] Li L, Lv Y, Wang F Y. Traffic signal timing via deep reinforcement learning [J]. IEEE/CAA Journal of Automatica Sinica, 2016, 3(3): 247 – 254.
- [6] Wei H, Zheng G J, Yao H X, et al. IntelliLight: A reinforcement learning approach for intelligent traffic light control [A]. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining [C]. London, UK: ACM, 2018. 2496 – 2505.
- [7] Wu L R, Zhang X R, Shi Z K. An intelligent fuzzy control for crossroads traffic light [A]. 2010 Second WRI Global Congress on Intelligent Systems [C]. Wuhan, China: IEEE, 2010. 28 – 32.
- [8] Dresner K, Stone P. Multiagent traffic management: a reservation-based intersection control mechanism [A]. Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems [C]. New York, USA: IEEE, 2004. 530 – 537.
- [9] Arnaud de La Fortelle. Analysis of reservation algorithms for cooperative planning at intersections [A]. 13th International IEEE Conference on Intelligent Transportation Systems [C]. Funchal, Portugal: IEEE, 2010. 445 – 449.
- [10] Perronnet F, Abbas-Turki A, Moudni A E. A sequenced-based protocol to manage autonomous vehicles at isolated intersections [A]. 16th International IEEE Conference on Intelligent Transportation Systems [C]. The Hague, Netherlands: IEEE, 2013. 1811 – 1816.
- [11] Xiao W, Cassandras C G. Decentralized optimal merging control for connected and automated vehicles [A]. Proceedings of 2019 American Control Conference [C]. Philadelphia, USA: IEEE, 2019. 4090 – 4095.
- [12] Zhang Y, Malikopoulos A, Cassandras C G. Optimal con-

trol and coordination of connected and automated vehicles at urban traffic intersections [A]. 2016 American Control Conference [C]. Boston, USA; IEEE, 2016. 6227 – 6232.

- [13] Zhang Y, Cassandras C G, Malikopoulos A. Optimal control of connected automated vehicles at urban traffic intersections; a feasibility enforcement analysis [A]. 2017 American Control Conference [C]. Seattle, USA; IEEE, 2017. 3548 – 3553.
- [14] Zhang Y, Cassandras C G. A decentralized optimal control framework for connected automated vehicles at urban intersections with dynamic resequencing [A]. 2018 IEEE Conference on Decision and Control [C]. Miami Beach, USA; IEEE, 2018. 217 – 222.
- [15] Fayazi S A, Vahidi A. Vehicle-in-the-loop (VIL) verification of a smart city intersection control scheme for autonomous vehicles [A]. 2017 IEEE Conference on Control Technology and Applications [C]. Mauna Lani, USA; IEEE, 2017. 1575 – 1580.
- [16] Qian X, Gregoire J, Moutarde F, et al. Priority-based coordination of autonomous and legacy vehicles at intersection [A]. 17th International IEEE Conference on Intelligent Transportation Systems (ITSC) [C]. Qingdao, China; IEEE, 2014. 1166 – 1171.

作者简介



吕 品 男, 1983 年 12 月出生, 山东滨州市人. 博士, 中国计算机学会高级会员. 现为广西大学计算机与电子信息学院副研究员, 硕士生导师. 主要研究领域为无线网络、物联网、人工智能等.

E-mail: lvpin@gxu.edu.cn



何岳滨 男, 1997 年 6 月出生, 福建厦门人. 现为广西大学计算机与电子信息学院硕士研究生. 主要研究领域为物联网、人工智能等.

E-mail: hyb051924@outlook.com



许 嘉 (通信作者) 女, 1984 年 4 月出生, 山东荣成人. 博士, 中国计算机学会高级会员. 现为广西大学计算机与电子信息学院副教授, 硕士生导师. 主要研究领域为大数据、人工智能等.

E-mail: xujia@gxu.edu.cn