多智能体运动规划在交通系统中的应用

彭泳谦 2100017731　　尹思源 2100017768

I. 引言

在现代城市环境中，高效的交通管理对于缓解拥堵和确保安全至关重要。在传统方法中，每个个体拥有一套独立的驾驶规则，在交通灯的协调之下，尽可能地防止碰撞。然而，随着交通密度的增加，这些方法往往无法满足效率和安全的要求。有研究表明，人类驾驶员停停走走的驾驶习惯和串联失稳的效应，会导致交通系统的效率与安全性变低 [1] [2]。近来，自动驾驶汽车和 5G 通信技术的快速发展，使利用算法控制自动驾驶汽车，以缓解交通拥堵、能源损耗成为可能。有研究表明，在模拟的环境中优化对于自动驾驶汽车的控制可以有效地减轻交通网络中的拥挤 [3] [4]。

本文提出了一种多机器人协同运动规划的方法，以优化交通效率，取消传统交通信号灯的需求。我们利用万物互联（IoE）的概念，提出了一种统一且高效的交通规划系统。我们采用了强化学习算法——深度确定性策略梯度（DDPG）[5]——以实现多机器人之间的实时自适应协调，提高交通路口的通行效率和安全性。具体来说，我们设计了一个虚拟交通环境，以一定的分布生成东西走向和南北走向的汽车，探究传统红绿灯控制与多智能体控制算法对于路口通行效率的影响。我们在此虚拟环境中，构建了一个中心化的强化学习问题：状态空间为当前时刻每一车辆的位置、速度和运动方向；动作空间为当前时刻每一车辆的加速度。我们利用 DDPG 算法 [5]，在此设定下对所有车辆进行一个中心化的控制，使所有车辆安全、快速地通过路口。这个任务十分具有挑战性，首先相比传统的多智能体问题，在我们的设定中，多智能体的数量很多并且在不停地发生变化，为了能够最大程度地利用全局的信息，我们采用了中心化的算法，这意味着状态空间和动作空间的维度都很高，这个特性使强化学习算法的训练变得十分困难。

通过设计多种指标，我们对该方法的性能进行了评估。结果表明，协同规划方法能够避免人类驾驶员停停走走的驾驶习惯，同时省去了一些不必要的等待红绿灯的时间，有效地提高了路口的通行效率。通过算法学习到的策略颠覆了人对于交通系统的传统认知，引起我们对于如何使用人工智能算法优化交通系统的思考。我们的代码开源在 *https://github.com/PPYYQQ/2024Spring-Robotics*，我们的视频 Demo 上传在 *https://PPYYQQ.github.io*。

II. 相关工作

A. 强化学习

强化学习（Reinforcement Learning）[6] 是一种通过与环境互动来学习最优决策策略的机器学习方法。与监督学习不同，强化学习不依赖于预先标注的数据，而是通过试错和反馈来优化策略。在强化学习中，智能体通过在环境中采取动作，从而影响环境状态，并根据正面或负面奖励来调整策略，以最大化长期累积奖励。强化学习的基本组成部分包括：

- 环境：智能体互动的对象和场景，包含随着时间变化的外部条件。
- 状态 $s \in \mathcal{S}$：环境在某一时刻的具体描述。
- 动作 $a \in \mathcal{A}$：智能体在特定状态下所采取的操作。
- 奖励 $R(s_t, a_t, s_{t+1})$：智能体采取某一动作后从环境中获得的反馈。
- 策略 π：智能体在各个状态下选择动作的规则或模型。
- 价值函数 $Q^\pi(s_t, a_t)$：评估每个状态或状态-动作对的长期累积奖励的期望值。

强化学习主要分为价值函数学习和策略梯度方法两种学习思路。随着深度神经网络的快速发展，这两种途径更加紧密地融合在一起，出现了演员-评论家、深度确定性策略梯度（DDPG）、信赖域策略优化和近端策略优化等方法。本文使用的是 DDPG 方法，在此对前两者作简要介绍。

1) 演员-评论家方法： 演员-评论家方法是一种强化学习框架，它结合了策略梯度方法和价值函数逼近

的方法。这个框架中包括演员和评论家两个主要组件。演员的任务是基于当前的策略生成动作。在每一个时间步，演员网络根据当前的状态输出一个具体的动作，即本文中每辆小车的加速度。评论家的任务是评估演员的动作是否良好。评论家通过估计状态-动作值函数（Q 值）来对演员的动作进行打分。Q 值是给定状态和动作对未来累计回报的期望值: $Q^{\pi}(s_t, a_t) = \mathbb{E}_{\pi}[r_t + \gamma Q^{\pi}(s_{t+1}, \pi(s_{t+1})]$.

演员-评论家方法的优势在于它将策略优化和价值评估分开进行，这样可以提高学习的稳定性和效率。演员网络直接优化策略，而评论家网络则通过降低策略的方差来引导演员更好地进行策略更新。

2) DDPG: 深度确定性策略梯度（DDPG）是一种用于连续动作空间中策略学习的算法。它使用了演员-评论家方法的框架，通过深度神经网络来近似策略和价值函数。与传统的 Q 学习主要用于离散动作空间不同，DDPG 能够处理连续动作，在本文工作我们使用该方法来控制场景中每辆车的加速度，以此实现运动规划任务。DDPG 采用经验回放和目标网络来稳定训练过程，其中经验回放涉及存储和重复利用过去的经验以打破连续更新之间的相关性，而目标网络通过提供更稳定的目标来减少更新的方差。算法的实现过程如 Algorithm 1 所示。

Algorithm 1 DDPG 算法伪代码

随机初始化 Critic 网络 $Q(s, a|\theta^Q)$ 和 Actor 网络 $\pi(s|\theta^{\pi})$
初始化目标网络权重参数 Q' 和 π'
初始化经验回放区 R
for episode = 1, M **do**
  行动探索，随机噪声 σ 初始化，$\sigma_i \sim N(0, 0.04)$
  获得初始观察状态 s_1
  for t=1, T **do**
    $a_t = \pi(s_t|\theta^{\pi}) + \sigma_t$
    执行动作 a_t，得到奖励 r_t 和环境状态 s_{t+1} 数据 (s_t, a_t, r_t, s_{t+1}) 存入 R
    从 R 中随机采样批量数目 N 的多维数组 (s_i, a_i, r_i, s_{i+1})
    $y_i = r_i + \gamma Q'(s_{i+1}, \pi'(s_{i+1}|\theta^{\pi'})|\theta^{Q'})$
    最小化均方损失函数 L 来更新 Critic 网络:

$$L = \frac{1}{N}\sum_i (y_i - Q(s_i, a_i|\theta^Q))^2$$

    采样策略梯度更新 Actor 策略网络:

$$\nabla_{\theta^{\pi}} J(\theta^{\pi}) \approx \frac{1}{N}\sum_i \nabla_{\theta^{\pi}} \pi(s_i|\theta^{\pi}) \nabla_a Q(s_i, a|\theta^Q)|_{a=\pi(s_i)}$$

    使用软更新机制更新目标网络:

$$\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}, \theta^{\pi'} \leftarrow \tau\theta^{\pi} + (1-\tau)\theta^{\pi'}$$

  end for
end for

B. 强化学习在交通系统中的应用

强化学习在交通系统中的应用具有巨大潜力，是一个非常有前景的发展方向。强化学习可以应用在交通信号控制、自动驾驶、车辆路线优化、多智能体协调等方面。利用强化学习算法，交通系统可以变得更加智能和高效。基于演员-评论家方法的多智能体强化学习是一个新兴领域。Aslani 等人提出了一种用于多个交叉口的连续空间演员-评论家控制模型 [7]，引入了新的编码和估值方式，并将模型在德黑兰市进行了测试。在另一项工作中，Abdoos 研究了一种两层层次结构的多智能体强化学习方法 [8]，该方法在每个路口使用基于 Q 学习的单个智能体，并在第二层使用基于平铺编码的函数近似器来控制广域网络。另外也有不少研究提出了邻近智能体之间的协调，以实现总体上的最佳表现。Tantawy 等人提出了一种基于 Q 学习的多智能体强化学习方法，用于道路网络协调 [9] [10]。他们展示了一个小规模的道路网络，并研究了多伦多市中心 59 个交叉口的大型网络。

我们期待未来城市的交通系统的智能化场景：自动驾驶汽车根据规划的路线，通过中心化的控制来协调多辆汽车之间的运动状态，动态调整路口的通行策略等等。但是现实世界的部署与使用学习算法的模拟器应用之间仍然存在巨大差距，例如安全性问题、对噪声的鲁棒性等等 [11]。这些问题仍需要相关领域的研究者艰苦卓绝的努力。

III. 问题建模

A. 符号定义

N 表示车辆的总数。

$\mathbf{s}_t$ 表示在时刻 t 的状态向量，包含所有车辆的位置、速度和运动方向：$\mathbf{s}_t = \left[\mathbf{s}_t^1, \mathbf{s}_t^2, \ldots, \mathbf{s}_t^N\right]$ 其中，$\mathbf{s}_t^i = (x_t^i, y_t^i, v_t^i, d_t^i)$ 表示第 i 辆车在时刻 t 的状态，包括位置 (x_t^i, y_t^i)、速度 v_t^i 和运动方向 d_t^i。

$\mathbf{a}_t$ 表示在时刻 t 的动作向量，包含所有车辆的加速度：$\mathbf{a}_t = \left[a_t^1, a_t^2, \ldots, a_t^N\right]$ 其中，a_t^i 表示第 i 辆车在时刻 t 的加速度。

B. 强化学习建模

1) 状态空间： $\mathcal{S}$ 定义为所有车辆在当前时刻的位置、速度和运动方向的组合：

$$\mathcal{S} = \left\{\mathbf{s} \in \mathbb{R}^{4N} \mid \mathbf{s} = \left[\mathbf{s}^1, \mathbf{s}^2, \ldots, \mathbf{s}^N\right], \mathbf{s}^i = (x^i, y^i, v^i, d^i)\right\}$$

2) 动作空间： $\mathcal{A}$ 定义为所有车辆在当前时刻的加速度组合：$\mathcal{A} = \left\{\mathbf{a} \in \mathbb{R}^N \mid \mathbf{a} = \left[a^1, a^2, \ldots, a^N\right]\right\}$

3) 奖励函数：

$$r_t = \sum_{i=1}^{N} \left(-10z(v_t^i) + 3v_t^i - 1000col_i + 500pass_i\right)$$

其中 $z(v_t^i)$ 在第 i 辆车速度为 0 时为 0，大于 0 时为 1；col_i 在第 i 辆车碰撞时为 1，不碰撞时为 0；$pass_i$ 在第 i 辆车顺利通过场景时为 1，否则为 0。

IV. 主要方法

A. 虚拟环境

我们构建了一个十字路口的环境，为了简化问题，车辆只有从西向东和从北向南的走向。车辆拥有固定的参数，包括：运动方向，最大速度和加速度范围。我们逐帧地对整个环境进行模拟，车辆以可调的分布从十字路口的最西侧或最北侧出现，运动到最东侧或最南侧则视为到达，到达的、在过程中发生碰撞的和由于拥堵而添加失败的车辆都会被立刻移出环境。为了交通灯系统和无交通灯系统能够进行对比，我们选取了同一个车辆出现的分布进行比较。

B. 交通灯系统

在使用交通灯信号控制的设定中，两个交通灯分别控制东西走向和南北走向的车辆，信号灯的时长可以进行调整。对于每一辆车，我们使用了 rule-base 的方法对车辆进行控制。具体来说，当车辆未到路口时，一直以最大的加速度进行运动直到达到最大速度。另外，为了避免碰撞车辆之间设有一个安全距离，当与前车的距离小于安全距离时，车辆就以最大的刹车减速度进行减速；当车辆在距离路口到一定距离的时候，就会根据信号灯的状态作出反应，绿灯时正常通行，红灯和黄灯时以最大的刹车减速度进行减速。为了获得一个比较强的 baseline，我们针对固定分布对信号灯转换策略进行了调节。

C. 无交通灯系统

在不使用交通灯信号控制的设定中，我们采用中心化的算法在每一时刻对所有车辆的加速度进行控制。具体来说，算法的输入是当前时刻的环境状态——所有车辆在当前时刻的位置、速度和运动方向的组合：

$$\mathcal{S} = \left\{\mathbf{s} \in \mathbb{R}^{4N} \mid \mathbf{s} = \left[\mathbf{s}^1, \mathbf{s}^2, \ldots, \mathbf{s}^N\right], \mathbf{s}^i = (x^i, y^i, v^i, d^i)\right\}$$

输出是所有车辆在当前时刻的加速度：

$$\mathbf{a}_t = \left[a_t^1, a_t^2, \ldots, a_t^N\right]$$

1) 模型： 我们使用了 DDPG 算法，采用了 Actor-Critic 的网络架构。其中 Actor 网络由带有激活函数的三个全连接层构成，隐藏神经元数量为 256、256。前两层的激活函数使用的是 Relu，最后一层的激活函数使用的是双曲正切函数，网络的输出是 [-1,1]，再通过线性映射映射到加速度范围的区间中。Critic 网络由四个全连接层构成，隐藏神经元数量为 400、400、32，其中前三层都带有 Relu 激活函数，网络的输出是一个标量，表示对当前状态和动作的联合评价。

2) 奖励函数：

$$r_t = \sum_{i=1}^{N} \left(-10z(v_t^i) + 3v_t^i - 1000col_i + 500pass_i\right)$$

奖励函数的设计，对于强化学习模型的训练至关重要。在此场景中，奖励函数主要由两个部分组成。奖励函数的第一个部分是稠密的，在每一帧都依据车辆的速度计算奖励，在模型训练和收敛的稳定性上起到了很重要的作用。具体来说，我们对完全静止的车辆进行了惩罚，并按比例对车辆的速度进行奖励。奖励函数的第二个部分是稀疏的，在发生一些关键事件的时候，给予数值较大的惩罚或奖励。具体来说，我们对由于拥堵导致的无法生成车辆和车辆碰撞进行了惩罚，对车辆成功到达终点进行了奖励。

D. 评价指标

- $Delay_t = \frac{\sum_{t=1}^{T} \#Cars_t}{T}$，其中 $\#Cars_t$ 表示 t 时刻在场景中的车辆的数量。这个指标描述了所有车辆出现在场景中的时长，在车辆出现分布相同的情况下，可以用来反映拥堵情况。
- $Flow_t = \frac{\#Passed_t}{t}$，其中 $\#Passed_t$ 表示截至时刻 t 总共到达终点的车辆的数量。这个指标描述了场景中车辆通过的速率。
- $AddFail_t$，表示截至时刻 t 由于拥堵而导致的车辆生成失败的次数。
- $Collision_t$，表示截至时刻 t 发生碰撞的车辆数量。这个指标反映了算法控制的交通系统的安全性。

V. 实验

A. 训练过程

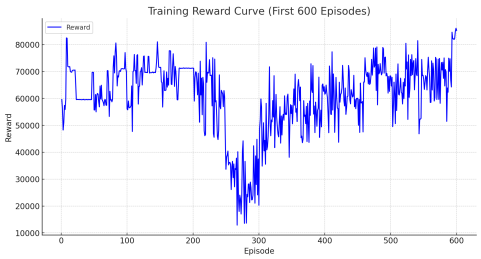
考虑到强化学习本身训练的不稳定性，而且状态空间和动作空间的维度较高，本文中采用的网络相对简单。我们在该环境中训练了 1500 个轮次，每个轮次有 1000 帧，在训练过程中，我们维护了一个最大容量为 100000 的经验池队列，也就是说每次训练网络只从最近 100 个轮次的记忆中抽取出一个批次对参数进行进行梯度下降和软机制更新。训练过程中奖励在随着轮次增加的变化曲线如图 1 所示。初始阶段的奖励值波动较大，部分集数的奖励值较低，甚至出现负值。这表明智能体在初期探索环境时尝试了许多不同的策略，未能立即找到有效的解决方案，随着轮次增加，奖励值逐渐上升，波动逐渐减少，奖励值总体维持在一个较高的水平。训练过程比较曲折，我们认为这可能和奖励设置相对稠密有关：一个动作的波动便可能引起奖励的巨大变动。但最终的奖励值基本能够稳定在 100000 左右。

图 1: 训练过程中奖励随着轮次增加的变化曲线

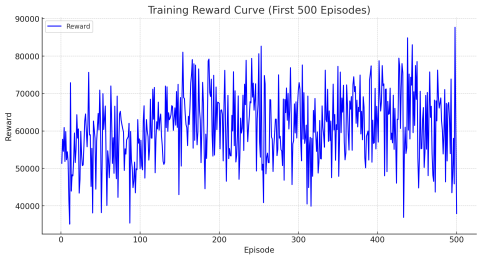
随后我们在以上训练的基础上改变了环境中的部分条件，将训练好的网络在新的环境中再次进行训练。我们尝试改变了以下条件：

- 小车出现的频率：即道路上每多少帧有可能出现一辆小车，原条件为纵向每 40 帧出现一辆小车，横向每 100 帧出现一辆小车。修改之后的条件为纵向每 50 帧出现一辆小车，横向每 80 帧出现一辆小车。
- 小车出现的概率：即每到一定帧率时决定该时刻是否出现小车。原条件为每隔一定帧率小车以概率 1 出现，修改之后的条件为小车以 0.75 的概率出现。

根据曲线变化我们发现，改变频率之后重新进行训练之后网络不能很好的适应新环境，而改变概率对网络没有造成很大的影响，奖励在一个较大的值附近波动。



(a) 改变小车出现的频率



(b) 改变小车出现的概率

图 2: 改变环境条件之后继续训练的奖励曲线

B. 实验结果

以上三种环境设置下，网络在测试中取得的各指标表现如表 I 所示，表中还展示了交通灯条件下的各项指标。仿真模拟以视频 Demo 的形式上传在 *https://PPYYQQ.github.io*

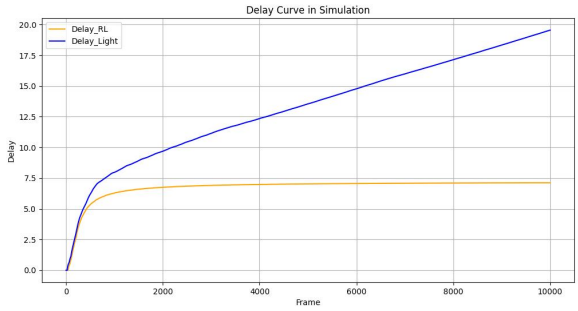
我们在——纵向每 40 帧出现一辆小车，横向每 100 帧出现一辆小车——的条件下对比了两个系统的性能。值得注意的是，由于在无交通灯的系统中，没有出现车辆生成失败的情况，为了反映有交通灯系统的车辆生成失败的情况，在模拟中，如果在生成车辆的位置处已经

环境条件	Delay	Flow	Add Fail	Collision
交通灯	16.571	23	-	0
未改变	**6.274**	**26**	0	0
改变频率	4.981	18	0	8
改变概率	6.288	**26**	0	0

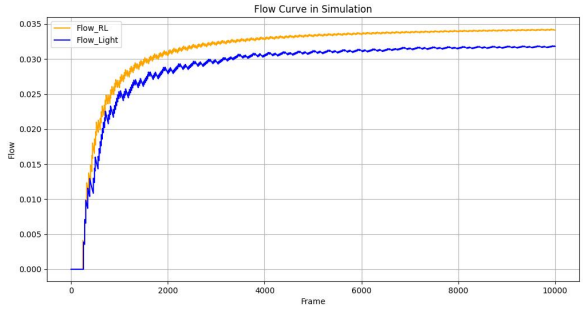
表 I: 不同环境条件下网络的各项指标表现

有车辆，依旧生成，以排队的方式进行处理，拥堵最终会体现在 Delay 曲线中。

评价指标如图 3 所示。在这种情况中，强化学习算法控制的无交通灯的系统表现良好，由于没有出现拥堵，Delay 曲线呈现出收敛的趋势。在有交通灯的系统中，由于车辆无法及时通过路口，导致生成的车辆不断堆积 Delay 呈线性增长，拥堵随着时间推移越来越严重。在 Flow-Frame 曲线图中也可以看出，无交通灯的系统有着比有交通灯系统更高的通行效率。可见无交通灯的系统，能够通过规划车辆的运动速度，在保证安全的情况下降低拥堵、提高通行效率。



(a) Frame-Delay 曲线



(b) Frame-Flow 曲线

图 3

VI. 结论

A. 方法和实验结果总结

我们设计了一个虚拟交通场景并提出 Delay 等指标来描述路口的通行效率。在此基础上我们利用强化学习的方法，通过 DDPG 算法训练得到了一个中心化的策略网络，对场景中所有车辆的加速度进行调控以实现对车辆速度的控制。相比于传统地通过红绿灯进行交通控制的方法，我们提出的方法有效地缓解了路口的拥堵、提高了通行的效率，以一个全新的视角看待交通系统，希望能对相关的研究有所启发。

B. 改进方向与未来工作

- **鲁棒性：** 我们训练完成的网络在改变小车出现的频率之后表现不佳，会出现小车相撞的情况。本方法尽管在通过加速度调控速度的时候添加了噪声，但依然对环境条件的改变较为敏感，鲁棒性有待提高。这与网络结构简单，训练不够充分有关。未来本方法应用在更复杂的场景甚至现实场景中，要增大网络和训练数据的规模，保持较高的性能，提高鲁棒性。
- **考虑实际场景：** 在本文中，我们对交通系统作了大量简化，只考虑了一个十字路口，小车进行加速和减速的情况。实际场景中还有多车道、多叉路口、高架桥等复杂情况，小车还有变道、转弯等更复杂的行为，以及各辆车之间的性质和性能不同。同时还应考虑实际驾驶场景中，人们对于舒适性的需要。针对此问题，奖励函数中还可以考虑速度稳定性、加速度的连续性等的优化。这些更加复杂的场景和符合实际应用需要的因素是我们未来努力的方向。
- **强化学习的其他算法：** 在本文中我们选取了 DDPG 算法来训练我们的网络，此外还有近端策略优化 (PPO)，SAC(soft-actor-critic) 可以尝试，这些算法已经在很多任务上被验证拥有更优越的性能，我们可以将更多的算法应用到本文提出的场景中，对模型进行改进。

参考文献

[1] Sugiyama, Y., Fukui, M., Kikuchi, M., Hasebe, K., Nakayama, A., Nishinari, K., ... & Yukawa, S. (2008). Traffic jams without bottlenecks—experimental evidence for the physical mechanism of the formation of a jam. New journal of physics, 10(3), 033001.

[2] Wu, C., Bayen, A. M., & Mehta, A. (2018, May). Stabilizing traffic with autonomous vehicles. In 2018 IEEE international conference on robotics and automation (ICRA) (pp. 6012-6018). IEEE.

[3] Wu, C., Kreidieh, A., Vinitsky, E., & Bayen, A. M. (2017, October). Emergent behaviors in mixed-autonomy traffic. In Conference on Robot Learning (pp. 398-407). PMLR.

[4] Vinitsky, E., Parvate, K., Kreidieh, A., Wu, C., & Bayen, A. (2018, November). Lagrangian control through deep-rl: Applications to bottleneck decongestion. In 2018 21st International Conference on Intelligent Transportation Systems (ITSC) (pp. 759-765). IEEE.

[5] Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., ... & Wierstra, D. (2015). Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971.

[6] "ROBOT LEARNING, edited by Jonathan H. Connell and Sridhar Mahadevan, Kluwer, Boston, 1993/1997, xii+240 pp., ISBN 0-7923-9365-1 " Robotica, vol. 17, no. 2, pp. 229 – 235, 1999. doi:10.1017/S0263574799271172

[7] M. Aslani, M. S. Mesgari and M. Wiering, "Adaptive traffic signal control with actor-critic methods in a real-world traffic network with different traffic disruption events", Transp. Res. C Emerg. Technol., vol. 85, pp. 732-752, Dec. 2017.

[8] M. Abdoos, N. Mozayani and A. L. C. Bazzan, "Hierarchical control of traffic signals using Q-learning with tile coding", Int. J. Speech Technol., vol. 40, no. 2, pp. 201-213, Mar. 2014.

[9] S. El-Tantawy and B. Abdulhai, "Multi-agent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC)", Proc. 15th Int. IEEE Conf. Intell. Transp. Syst., pp. 319-326, Sep. 2012.

[10] S. El-Tantawy, B. Abdulhai and H. Abdelgawad, "Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC): Methodology and large-scale application on downtown toronto", IEEE Trans. Intell. Transp. Syst., vol. 14, no. 3, pp. 1140-1150, Sep. 2013.

[11] A. Haydari and Y. Yılmaz, "Deep Reinforcement Learning for Intelligent Transportation Systems: A Survey," in IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 1, pp. 11-32, Jan. 2022, doi: 10.1109/TITS.2020.3008612.