

# 基于深度 Q 网络的地铁列车速度曲线 优化方法研究

孔文龙<sup>1</sup>, 柏友运<sup>1</sup>, 向国良<sup>1</sup>, 范子寅<sup>2,3</sup>

(1. 长沙穗城轨道交通有限公司, 长沙 410129;

2. 通号城市轨道交通技术有限公司, 北京 100070;

3. 中南大学 交通运输工程学院, 长沙 410075)

**摘要:** 随着国内地铁运营规模的不断扩大, 地铁列车电能消耗过大的问题亟待解决; 通过对列控系统速度曲线进行优化, 减小列车运行时因克服阻力做功和制动过程造成的机械能损失, 提高牵引能效; 在算法层面上, 充分利用 AI 算法和高性能计算设备在处理最优化问题方面的优势, 提出基于深度 Q 网络算法的速度曲线优化方法: 该方法应用强化学习架构, 通过对智能体进行迭代优化, 求解出具有节能特性的速度曲线; 在 Matlab 平台上搭建了仿真环境, 对传统算法和所提算法的节能效果进行了仿真对比, 结果显示, 所提算法相比传统算法实现了约 5%~7% 的节能。

**关键词:** 地铁列车; 牵引能效; 速度曲线; 深度 Q 网络; 强化学习架构

## Research on Speed Profile Optimization Method for Metro Trains Based on Deep Q-Network

KONG Wenlong<sup>1</sup>, BO Youyun<sup>1</sup>, XIANG Guoliang<sup>1</sup>, FAN Ziyin<sup>2,3</sup>

(1. Changsha Suicheng Rail Transit Co., Ltd., Changsha 410129, China;

2. Urban Rail Transit Technology Co., Ltd., CRSC, Beijing 100070, China;

3. School of Traffic and Transportation Engineering, Central South University, Changsha 410075, China)

**Abstract:** With the continuous expansion of domestic metro operations, it is of urgency to solve excessive energy consumption in metro trains. The speed profile of the train control system is optimized to reduce the mechanical energy loss caused by overcoming resistance and braking during train operation, thereby improving traction energy efficiency. At the algorithmic level, by fully leveraging the advantages of AI algorithms and high-performance computing equipment in handling optimization problems, a speed profile optimization method based on the deep Q-network algorithm is proposed, which employs a reinforcement learning framework and iteratively optimizes the agent to derive an energy-efficient speed profile. Through simulation environments on the MATLAB platform, a comparison in energy-saving between the traditional algorithms and the proposed algorithm is compared, which shows that the proposed algorithm has the advantage over traditional methods with the energy savings of about 5% to 7%.

**Keywords:** metro trains; traction energy efficiency; speed profile; deep Q-network; reinforcement learning architecture

## 0 引言

近五年来, 我国内地的地铁交通运营规模不断扩大, 地铁系统的总用电量和运营成本激增的问题也日益突出<sup>[1-5]</sup>。通过对列控系统的速度曲线进行优化, 合理

规划不同运行阶段的目标速度, 可以提升列车的牵引能效, 减少运行过程的电能消耗, 进而降低运营成本并减少环境污染, 促进可持续发展。这一技术不仅会对轨道交通运营企业带来巨大的经济效益, 也对社会的绿色出行和环境保护具有深远意义<sup>[6]</sup>。

收稿日期: 2025-06-09; 修回日期: 2025-09-11。

作者简介: 孔文龙(1975-), 男, 大学本科, 高级工程师。

通讯作者: 范子寅(1999-), 男, 硕士。

引用格式: 孔文龙, 柏友运, 向国良, 等. 基于深度 Q 网络的地铁列车速度曲线优化方法研究[J]. 计算机测量与控制, 2026, 34(2): 182-188, 265.

早期的速度曲线优化研究多采用最优控制理论, 如最大值原理、伪谱法等, 将列车运行曲线优化问题转化为最优控制问题求解。例如, 20 世纪六七十年代, 文献 [7] 在牵引效率不变、无坡道、无弯道、无限速等理想的假设条件下, 建立列车平直道运行模型, 利用最大值原则求解出列车最优速度曲线包括最大牵引、巡航、惰行及最大制动共 4 种工况。随后在 80 年代, 文献 [8] 在考虑固定坡道的情况下, 建立了以最小化列车机械能耗为目标的模型, 采用极大值原理求解列车运行工况。文献 [9] 将列车最大牵引力以及列车最大制动力作为限制条件进行考虑, 基于极大值原理求解定时间下的列车节能运行方案。文献 [10] 随后从数学上证明, 列车在区间内运行, 满足时间约束的条件下, 其行程中应包括一个加速相位、一个匀速相位、一个惰行相位以及一个制动相位, 同时给出了各相位下能耗的数学模型。在国内, 文献 [11] 同样采用极大值原理对列车速度曲线进行优化。

伪谱法的应用则相对更晚, 在最优控制领域得到了广泛应用。文献 [12] 等使用伪谱法将规划问题转化为多阶段最优控制问题, 然后转化为非线性规划问题进行研究。文献 [13] 采用伪谱法对自动驾驶的储能式有轨电车速度轨迹优化问题求解, 实现了有轨电车在全局求解性能上的最优。

传统优化方法主要适用于结构化问题, 具有明确的问题描述和条件, 通常依赖于梯度信息, 计算效率高, 但容易陷入局部最优, 且对初始值敏感。传统优化方法在处理复杂、多极值问题时存在局限性, 随着 AI 算法的发展, 诸如遗传算法 (GA)、粒子群优化 (PSO)、模拟退火 (SA)、蚁群优化 (ACO) 等众多方法均已在速度曲线优化方面得到了成功应用<sup>[14]</sup>。文献 [15] 提出了一种基于粒子群优化算法, 提出了一种结合城轨供电系统仿真平台和智能算法的优化方法。文献 [16] 将粒子群算法和遗传算法两种智能计算方法结合应用于城轨列车节能优化操纵曲线的设计上。文献 [17] 应用蚁群优化算法、遗传算法和动态规划对列车的速度曲线进行成功优化。此外还有文献 [18]、文献 [19] 等通过遗传算法, 对列车速度曲线的优化进行更深入的分析。

为了进一步发挥 AI 算法的优势, 本文提出了一种基于深度 Q 网络 (DQN, deep Q-network) 算法的速度曲线优化方法。通过对运行区间进行离散化处理、对动作空间进行设计和对参数变化范围进行设定, 实现了寻优框架的构建, 并在此基础上应用强化学习架构对智能体进行迭代优化, 最终优化得出具有节能特性的速度曲线。通过基于 Matlab 平台的仿真测试验证了所提算法的有效性。

## 1 速度曲线优化节能原理

地铁列车的运行能耗主要源于克服阻力做功和制动过程造成的机械能损失, 通过对列车自动驾驶 (ATO, automatic train operation) 系统的目标速度曲线进行优化, 可以避免部分不必要的牵引能量消耗, 实现能源利用效率的最大化。上述节能思路可通过以下 3 个方面加以实现:

1) 列车运行过程所受阻力可分为基本阻力和附加阻力两部分, 其中, 基本阻力包括由轴颈摩擦、滚动摩擦等原因引起的机械阻力和空气阻力。当列车速度较慢时, 机械阻力是基本阻力的主要部分, 这部分阻力的大小与列车速度呈线性关系; 当列车速度较快时, 空气阻力变为基本阻力的主导, 这部分阻力与列车速度呈非线性关系, 并且随速度的上升大幅增加。列车在高速阶段时, 为了减少输出的牵引功率, 同时维持较快的速度, 经常采用惰行的方式运行。若提高此阶段的目标速度, 可以使列车具有更长的惰行区间, 充分利用车身动能, 但同时也会受到更大的空气阻力, 增加克服阻力做功消耗的能量。因此, 需合理选择列车的目标速度, 减小机械能损失。

2) 列车运行阻力中的附加阻力包括列车通过坡道、曲线、隧道等路段产生的线路阻力, 这部分阻力对列车的动力学特性有显著影响, 因此, 进行速度曲线优化时必须考虑线路条件因素, 将坡度、曲线半径、隧道长度等信息引入优化流程。例如, 在平直轨道场景中, 采用“牵引—惰行—牵引”分段控制策略, 当列车加速至临界速度时, 关闭牵引电机进入惰行工况, 使列车仅克服基本阻力做功, 降低牵引能耗; 在连续上坡场景中, 采用“提前加速—恒速爬坡”策略, 在坡前平直段提前加速至目标速度, 利用动能储备抵消部分势能需求, 使爬坡阶段牵引功率需求降低; 在曲线密集区段, 采用“降速入弯”策略, 将轮轨侧向力引起的滚动阻力增幅控制在较小范围内, 同时减少因横向振动导致的乘客舒适度劣化。通过线路特性与列车性能的深度适配, 生成定制化的速度曲线, 使列车尽可能地利用线路势能, 并借助附加阻力代替部分制动力, 以更大程度地达到运行节能的效果。

3) 地铁列车运行时起制动频繁, 而制动阶段造成的牵引能量损失是制约系统能效提升的一大瓶颈, 因此, 必须从列车驾驶策略层面对制动的频次、时机等进行优化。例如, 在长大下坡道或进站减速场景中, 提前计算列车动能与势能的转换阈值, 通过“惰行—制动”协同控制策略, 使列车在速度衰减至临界值时再启动制动, 避免因过早制动导致不必要的动能损耗; 在连续上坡—下坡场景中, 可适当降低列车通过坡顶时的目标速

度，避免列车因下坡时速度过快而采取制动措施，造成坡道势能无法被充分利用的后果。此外，降低制动工况占比也有利于牵引节能，但需要注意防止列车超速和无法按时到站。

## 2 列车运行过程受力分析

如图 1 所示，列车运行方向的前向受到牵引力，后向受到基本阻力、附加阻力和制动力；列车因重力作用对轨道有一个法向的压迫力，并同时受到轨道的反向支撑作用力。根据铁路行业标准 TB/T 1407.1—2018《列车牵引计算》<sup>[20]</sup>，可以计算出其中几个关键的作用力，作为后续的速度曲线优化过程的环境变量。

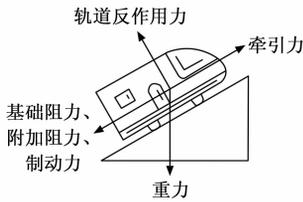


图 1 列车运行过程受力情况

基本阻力的计算采用戴维斯公式：

$$F_0 = A_0 + Mg(A + Bv) + Cv^2 \quad (1)$$

式中， $F_0$  为基本阻力， $M$  为列车车重， $g$  为重力加速度， $v$  为列车运行速度， $A_0$ 、 $A$ 、 $B$ 、 $C$  为基本阻力公式系数。

附加阻力是由线路条件引起的额外阻力，主要包括坡道阻力、曲线阻力和隧道阻力。其中，坡道阻力和曲线阻力通过下列经验公式计算得出：

$$\begin{cases} F_i = \frac{Mg \cdot i}{1000} \\ F_r = \frac{Mg \cdot r}{1000} \end{cases} \quad (2)$$

式中， $F_i$  为坡道阻力； $F_r$  为曲线阻力； $i$  是由坡段纵断面高差和坡段长度化简得到的坡度千分数，用以衡量坡段的倾斜程度； $r$  是由曲线计算长度和曲线半径化简得到的曲线折算坡度千分数，用以衡量曲线的弯曲程度。

$i$  和  $r$  的简化方式为：

$$\begin{cases} i = \frac{H_2 - H_1}{l_h} \times 1000 \\ r = \frac{600}{l_h} \sum \frac{l_r}{R} \end{cases} \quad (3)$$

式中， $H_1$ 、 $H_2$  为坡段纵断面两端标高 (m)， $l_h$  为化简路段长度 (m)， $l_r$  为曲线计算长度 (m)， $R$  为曲线半径 (m)。

隧道阻力通过下列经验公式计算得出：

$$F_s = \frac{Mg \cdot s}{1000} \quad (4)$$

式中， $F_s$  为曲线阻力， $s$  为隧道折算坡度千分数，由专门的实车试验测出。

列车的总运行阻力可视作基本阻力和附加阻力的叠

加，记作：

$$F_d = F_0 + F_i + F_r + F_s \quad (5)$$

式中， $F_d$  为列车运行阻力。

## 3 速度曲线优化设计

列车在相邻两站间运行时，根据其在路程中每个离散点处的目标速度，可以绘制出一条目标速度轨迹，即速度曲线。由于列车在每个离散点处都有无数种驾驶行为可做选择，即使对其运行时长进行了限制，理论上仍存在无数条满足要求的速度曲线。为了能在有限的时间和计算资源下完成寻优，现对速度曲线的优化方式进行如下设计。

### 3.1 运行区间离散化处理

设相邻两个站之间的路程为  $L$ 。如图 2 所示，将这段路程平均划分为  $n$  个子区间。在每个子区间的起始处，列车选择一种驾驶动作，并在该子区间中保持这一状态，直至子区间终止处。例如，列车在  $P_2$  位置选择“最大牵引”驾驶动作，则在第 2 子区间中列车将持续以最大牵引力加速行驶，到达  $P_3$  位置后再切换至下一驾驶动作。

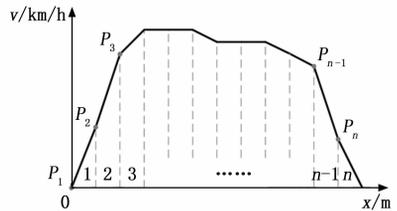


图 2 子区间划分示意图

子区间的数量  $n$  反映了离散化分的疏密程度，取不同的  $n$  值会对寻优过程有不同的影响。若  $n$  较小，则列车选择驾驶动作的机会较少，寻优范围相对受限；若  $n$  较大，则具有更大的寻优范围，但同时也提高了目标函数的收敛难度。本文取  $n=100$ 。

### 3.2 候选动作集设计

在每个子区间的起始处，列车可以选择的驾驶动作的集合就是候选动作集。本文采用的候选动作集包括“最大牵引”“巡航”“惰行”“最大制动”4 个驾驶动作，其中：“最大牵引”指采用最大牵引力进行加速，“巡航”指保持当前速度不变，“惰行”指不输出任何牵引力或制动力，“最大制动”指采用最大制动力进行减速。

### 3.3 参数变化范围限制

地铁列车实际运行时受到诸多来自设计标准、运行规范、安全手册等文件的限制。为使优化后的目标速度曲线满足运行要求，同时加快寻优速度，现对目标速度曲线的优化过程施加如下限制条件：

1) 根据 GB50157—2013《地铁设计规范》<sup>[21]</sup>，地铁列车的启动加速度不宜超过  $0.9 \text{ m/s}^2$ ，制动时的减速度

不宜超过  $0.9 \text{ m/s}^2$ 。因此, 规定列车在第 1 子区间只能选择“牵引”驾驶动作, 且对应的牵引力恒定为  $0.9 \text{ M}$ ; “最大制动”驾驶动作对应的制动力恒定为  $0.9 \text{ M}$ 。

2) 列车在站间运行时须严格遵守速度限制, 既不可超速行驶, 也不可在还未到站时停车或倒行。

3) 为保证列车停站后车门与站台门对齐, 业内普遍规定列车的站台停车精度为  $\pm 0.3 \text{ m}$ 。因此, 规定只有当目标速度曲线终端和  $v=0$  轴的交点与目标终点的距离小于等于  $0.3 \text{ m}$  时, 目标速度曲线才是有效的。

4) 为确保列车准点运行, 《城市轨道交通服务质量评价规范》<sup>[22]</sup>指出, 正点列车到达终到站的时刻与列车运行图计划时刻相比误差不大于  $2 \text{ min}$ 。为保证列车正点运行, 同时预留充足的停站冗余时间, 规定列车在相邻两站之间的运行时长与目标时长的偏差不得超过  $1 \text{ s}$ 。

5) 为了降低解空间复杂度, 规定列车在最后 4 个子区间(即第 97~100 子区间)只能选择“制动”驾驶动作。同时, 为了更容易实现精准停站和准时运行, 降低寻优难度, 规定列车在最后 4 个子区间可以在一定范围内输出任意的制动力。

## 4 速度曲线寻优算法

基于第 3 章的框架设计和限制条件, 采用 DQN 算法求解满足要求且具备节能特性的速度曲线。

### 4.1 强化学习架构

传统的 Q 学习存在泛化能力较差、样本效率低下、处理高维状态空间的能力有限等不足。为解决上述不足, 提高智能体处理问题的性能, DQN 采用了如图 3 所示的强化学习架构设计: 利用卷积神经网络(CNN, convolutional neural network)近似 Q 函数, 通过 CNN 的卷积层和池化层提取对象特征, 并利用全连接层将这些特征映射到相应的 Q 值, 极大地发挥了神经网络强大的表征能力和泛化能力, 可以更好地解决复杂环境下的决策问题。

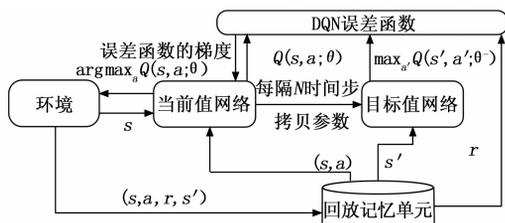


图 3 基于强化学习框架的 DQN 架构设计

### 4.2 经验回放机制

经验回放的基本思想是: 在算法执行过程中, 将所有的经验——即智能体在每个时间步采取的动作、观察到的奖励以及下一个状态, 以元组  $\{s, a, r, s'\}$  的形式——存储在一个经验回放池中。这个经验回放池具有固定的容量, 当新的经验加入时, 如果记忆已满, 则最早

的经验会被移除, 以此保证记忆的更新。

在训练深度 Q 网络时, 并非直接使用最新的经验进行更新, 而是从经验回放池中随机抽取一个小批量的经验。这种方式打破了训练样本之间的时间相关性, 避免了由于连续样本之间的相似性而导致的网络陷入局部最小值的问题。此外, 经验回放还使得训练任务更接近于通常的监督学习, 从而简化了程序的调试和算法的测试。具体来说, 在每次迭代中, 先从回放记忆中随机抽取一批经验, 再根据这些经验来计算损失函数。损失函数衡量了当前 Q 网络对动作值的估计与目标值之间的差异。目标值是根据经验中的奖励和下一个状态的最大 Q 值来计算的。然后, 使用梯度下降算法来更新 Q 网络的参数, 以最小化损失函数。通过这种方式, 经验回放不仅提高了训练的稳定性, 还使得智能体能够从过去的经验中学习更多有用的信息, 从而加快了学习的速度并提高了学习的效果。

### 4.3 “探索—利用”平衡策略

智能体的学习过程本质上是一种试错学习。在学习的初期, 智能体对环境及其工作机制并不清楚, 不知道在某一状态下哪种行为(动作)是正确的, 哪种是错误的。因此, 智能体需要从自身的经验中探索出一种好的策略, 以在寻优过程中获得更多的奖励。

在寻优过程中, 存在着探索(Exploration)和利用(Exploitation)之间的权衡。探索意味着智能体放弃一些已知的信息, 采用随机的方式进行当前动作的选择, 而利用则是指智能体根据已知信息采取最大化奖励的动作。探索和利用是深度强化学习中一个核心且复杂的问题。一方面, 智能体需要通过探索来发现新的、可能带来更高奖励的行为和策略; 另一方面, 智能体也需要利用已知的信息来优化当前的行为, 以确保在每一次决策中都能获得尽可能高的奖励。这种权衡是深度强化学习算法设计中的一个关键挑战, 它直接影响到智能体的学习效率 and 最终性能。

在 DQN 中, 探索和利用的权衡通常通过  $\epsilon$ -贪心策略( $\epsilon$ -greedy policy)来实现。在这个策略中, 智能体可能以  $\epsilon$  的概率随机选择一个动作进行探索, 也可能以  $1-\epsilon$  的概率选择当前估计的具有最高 Q 值的动作进行利用。这种策略允许智能体在探索新策略和利用已知最优策略之间进行平衡, 从而在不断的学习过程中逐步优化其行为。

### 4.4 更新迭代流程

为了用具有参数  $\theta$  的模型近似替代 Q 学习的价值函数  $Q(s, a; \theta)$ , DQN 引入了深度卷积神经网络。在每次时间步长  $t$  进行经验回放时, 智能体的经验  $e_t = (s_t, a_t, r_t, s_{t+1})$  被存储到数据集  $D_t = \{e_1, \dots, e_t\}$  中。

在第  $i$  次迭代时, DQN 采用以下损失函数进行更新:

$$L_i(\theta_i) = E_{(s_i, a_i, r_i, s'_i)} [\{y_i - Q(s, a; \theta_i)\}^2] \quad (6)$$

其中:

$$y_i = r + \gamma_{\max} Q(s', a'; \theta_i) \quad (7)$$

式中,  $\theta_i$  是在第  $i$  次迭代时用于计算目标 Q 网络的参数。

通过建立具有参数  $\theta$  的深度卷积神经网络, 可以近似估计最优状态—动作价值函数  $Q^*(s, a)$ , 即:

$$Q(s, a; \theta) \approx Q^*(s, a) \quad (8)$$

将损失函数对第  $i$  次迭代时的参数  $\theta_i$  求导, 得到梯度:

$$\nabla_{\theta_i} L_i(\theta_i) = E_{(s_i, a_i, r_i, s'_i)} [R_Q \nabla_{\theta_i} Q(s, a; \theta_i)] \quad (9)$$

式中,  $R_Q$  为:

$$R_Q = r + \gamma_{\max} Q(s', a'; \theta_i) - Q(s, a; \theta_i) \quad (10)$$

使用梯度下降的方法更新 Q 网络的参数  $\theta$ , 以最小化损失函数。历经多次迭代后, 智能体将学会在给定状态下选择能够最大化长期回报的动作。

根据图 3 所示的算法框架, DQN 训练过程包括以下几个步骤:

- 1) 初始化: 初始化 Q 网络和目标 Q 网络的参数  $\theta$  和  $\theta^-$ 。通常, 目标 Q 网络的初始参数与 Q 网络相同。
- 2) 经验回放: 从经验回放缓冲区中采样一批经验 (包括状态、动作、奖励、下一状态)。
- 3) 计算目标 Q 值: 对于采样的每个经验, 计算相应的目标 Q 值。
- 4) 计算损失: 使用当前 Q 网络预测 Q 值, 并计算与目标 Q 值之间的均方误差损失。
- 5) 反向传播: 计算损失函数关于网络参数的梯度。
- 6) 参数更新: 使用梯度下降法更新 Q 网络的参数。
- 7) 目标网络更新: 每隔一定的时间步长, 将 Q 网络的参数复制到目标 Q 网络, 即  $\theta^- \leftarrow \theta$ 。

DQN 的性能高度依赖于更新迭代过程中的关键参数设定, 例如折扣因子和学习率。这些参数的合理选择与优化是确保模型有效收敛并找到最优节能策略的根本保障。

折扣因子  $\gamma$  的选择直接影响智能体对未来奖励的评估, 其数值大小决定了智能体在进行决策时是更注重即时奖励还是长期回报。一个较小的  $\gamma$  会使智能体变得“短视”, 倾向于选择能立即带来高奖励的动作, 但这在列车运行这类需要全局规划的任务中, 可能导致策略陷入局部最优, 例如因前期过度加速而造成后期不必要的紧急制动, 反而增加了总能耗。相反, 一个较大的  $\gamma$  则使智能体具备“远见”, 会更全面地考虑当前决策对整个运行区间总能耗的影响。考虑到列车速度曲线优化的目标是 minimized 全程的总牵引能耗, 而非某一区间的瞬时能耗, 因此需要智能体具备较长的决策视野。本文经过初步实验与分析, 最终选取  $\gamma = 0.92$ 。这一取值在保

证智能体足够重视长期回报的同时, 也避免了因  $\gamma$  过分接近 1 而可能导致的价值函数过大、训练不稳定或收敛缓慢的问题, 在长期目标与训练效率之间取得了良好平衡。

学习率  $\alpha$  决定了模型在每次迭代中更新网络权重的幅度。 $\alpha$  的设定是一个经典的难题: 若  $\alpha$  过大, 可能导致损失函数在最优解附近震荡, 难以收敛; 若  $\alpha$  过小, 则会使模型收敛速度极为缓慢, 且容易陷入局部极小值。为解决这一问题, 本文没有采用固定的  $\alpha$ , 而是选用了自适应动量梯度下降法作为神经网络的训练算法。该方法融合了两种优化思想: 1) 它能够根据训练过程中损失函数的变化趋势自适应地调整  $\alpha$ , 从而在训练初期使用较大的  $\alpha$  以加快收敛, 在接近最优解时减小  $\alpha$  以实现精细调整; 2) 它引入了动量机制, 通过累积历史梯度信息来平滑更新方向, 帮助优化过程“冲出”鞍点或局部极小值的“洼地”, 加速收敛并提高模型的泛化能力。通过采用这种自适应优化器, 不仅避免了手动调节  $\alpha$  的繁琐过程, 还显著提升了 DQN 模型在复杂非凸优化问题中的训练稳定性和寻优效率。

#### 4.5 奖励函数

在强化学习中, 奖励是智能体从环境中获得的反馈信号, 用于评估其行为的优劣。奖励引导智能体学习, 通过不断尝试和调整策略, 智能体旨在最大化长期累积奖励。基于“稀疏奖励”和“轻奖重罚”的原则, 本文的奖励函数设计如下:

- 1) 在当前子区间终止处, 若本轮寻优既没有因违反约束条件而中断, 也没有随着到达终点而结束, 则当前子区间的奖励为 0, 即:

$$r = 0 \quad (11)$$

- 2) 在第  $n$  子区间, 若列车速度小于等于 0, 则立即结束本轮寻优, 获取奖励。

$$r = -50 \quad (12)$$

- 3) 在第  $n$  子区间, 在列车速度大于 0 的前提下, 若列车速度大于线路限速, 或者列车失去准时到站的可能性, 则立即结束本轮寻优, 获取奖励。

$$r = -50 + 0.1 \cdot n \quad (13)$$

- 4) 在第 97 子区间起始处, 若列车速度小于等于线路限速, 但是大于精准停站的速度上限  $v_{\max}$ , 则立即结束本轮寻优, 获取奖励。

$$r = -30 - 10 \cdot \frac{v(x) \Big|_{x=96 \cdot L/100} - v_{\max}}{v_{\text{limit}}(x) \Big|_{x=96 \cdot L/100} - v_{\max}} \quad (14)$$

- 5) 完成最后 4 个子区间的制动比例寻优后, 若寻优结果的“准时运行”适应度和“精准停站”适应度没有同时大于 0, 则获取奖励。

$$r = -10 \quad (15)$$

- 6) 完成全部子区间的驾驶动作决策后, 若得到的

目标速度曲线同时满足“准时运行”和“精准停站”要求, 则获取奖励。

$$r = 10 + 20 \cdot \frac{E_s - E}{E_s} \quad (16)$$

式中,  $E$  为列车在站间运行的总牵引能耗;  $E_s$  为列车节能奖励的计算标准值。

7) 除了上文提到的情形外, 其他情况下奖励均按 0 进行结算。

### 5 仿真测试

为验证所提速度曲线优化算法的有效性, 现基于某地铁线路相邻两站间的线路信息和某 A 型六编组地铁列车的特性参数, 在 Matlab 平台上搭建仿真环境, 对不同运行时长下的列车能耗优化过程进行仿真实验, 并与基于强化学习的传统速度曲线优化算法<sup>[23]</sup>进行对比。

#### 5.1 固定运行时长场景优化对比

设置列车的载荷状态为 AW2, 目标运行时长为 158 s, 在同一线路下进行测试, 此时传统算法和本文算法的优化结果如下。

图 4 展示了两种算法优化后的速度曲线。这两条速度曲线均在首段和末段进行了持续的大幅加速和制动, 但在中间部分做了不同的优化。从驾驶策略上看, 传统算法倾向于在中间段保持一个相对较低的速度, 并在较晚的时机开始惰行降速和制动减速; 本文算法则采取“先快后慢”的策略, 先将速度提升至接近限速, 维持一段路程后再开始惰行降速, 充分利用列车存储的动能和线路势能向前行进。

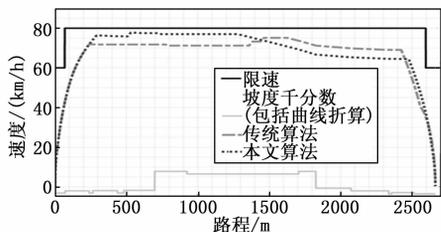


图 4 158 s 运行时长下的速度曲线对比

图 5 和图 6 对比了两种算法的巡航和惰行策略。由图可知, 相比传统算法, 本文算法减少了巡航工况的频次, 增加了惰行工况的频次。通过改变巡航时的目标速度和惰行时机, 本文算法将传统算法优化得到的部分巡航区段替换为惰行区段, 增加了惰行时长和距离, 更加

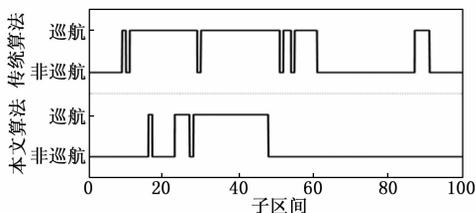


图 5 158 s 运行时长下的巡航策略对比

充分地利用了列车存储的机械能。

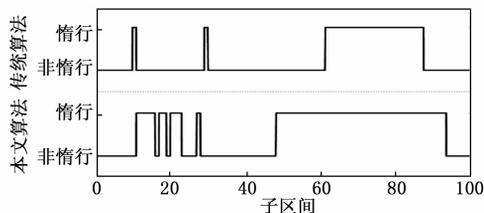


图 6 158 s 运行时长下的惰行策略对比

表 1 汇总了传统算法与本文算法优化结果的牵引能耗。由表可知, 传统算法求出的速度曲线具有更高的牵引能耗, 而本文算法优化得到的速度曲线降低了 6.95% 的牵引能耗, 展现出了显著的节能效果。

表 1 158 s 运行时长下的牵引能耗对比

算法	牵引能耗/kWh	节能比例/%
传统算法	34.350	6.95
本文算法	31.964	

本文算法采用了  $\epsilon$ -贪心策略来实现寻优过程中的“探索-利用”平衡。为探究所引入的随机概率  $\epsilon$  是否会对算法的稳定性造成负面影响, 现基于当前固定运行时长场景对算法进行多次独立运行测试, 获得结果如表 2 所示。

表 2 158 s 运行时长下本文算法稳定性测试结果

运行次序	牵引能耗/kWh	节能比例/%
1	31.964	6.95
2	32.183	6.31
3	32.026	6.77
4	31.990	6.87
5	32.281	6.02
6	32.019	6.79

根据表 2 所示结果, 经过 6 轮独立运行测试, 本文算法求解速度曲线的节能比例总是在 6%~7% 范围内变化, 6 个数据的标准差约为 0.37%, 数据的相对波动幅度约为平均值的 5.6%, 总体偏于稳定, 说明  $\epsilon$  未对算法的稳定性造成显著影响。

#### 5.2 不同运行时长场景优化对比

设置列车的载荷状态为 AW2, 目标运行时长分别为 158、163、168 s, 在同一线路下进行测试, 相应的传统算法和本文算法的优化结果如下。

图 7 和图 8 展示了 158、163、168 s 三个运行时长场景下传统算法和本文算法优化得到的速度曲线。随着目标运行时间的加长, 传统算法采取了中间段整体降速的方式来调整运行耗时, 每增加 5 s 的目标运行时长, 中间段的平均速度下降约 3 km/h。本文所提算法则通过改变惰行起始时机来调整运行耗时。在前段速度基本不变的情况下, 通过提早进行长距离的惰行,

使列车更早地进入缓慢降速的阶段，以此增加后段的运行耗时。每增加 5 s 的目标运行时长，惰行起始位置延后约 200 m。

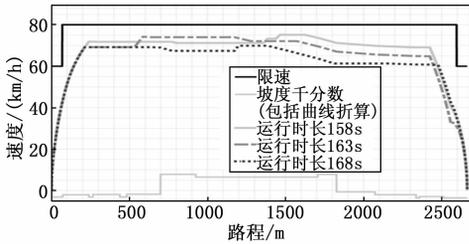


图 7 三个运行时长下传统算法优化得的速度曲线

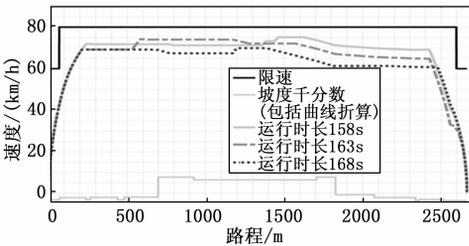


图 8 三个运行时长下本文算法优化得的速度曲线

表 3 汇总了 3 个运行时长场景下传统算法与本文算法优化结果的牵引能耗。在所有的运行时长下，相比传统算法，本文算法均降低了牵引能耗，节能比例达 5% 以上，验证了本文算法的节能特性。

表 3 三个运行时长下的牵引能耗对比

运行时长/s	算法	牵引能耗/kWh	节能比例/%
158	传统算法	34.350	6.95
	本文算法	31.964	
163	传统算法	31.684	5.47
	本文算法	29.951	
168	传统算法	29.863	7.10
	本文算法	27.744	

### 5.3 不同线路和车型场景优化对比

设置列车的载荷状态为 AW2，在线路 1 和线路 2 上分别对 A 型 6 编组地铁列车和 B 型 6 编组地铁列车的节能效果进行测试，结果如下。

表 4 汇总了 4 个测试案例中传统算法与本文算法优化结果的牵引能耗。结果显示，所有案例均实现了 5% 以上的牵引节能，验证了本文算法的普适性和鲁棒性。

## 6 结束语

AI 算法和计算机设备的快速发展使得数学领域的前沿成果得以应用于轨道交通行业。本文在此背景下提出了基于 DQN 的速度曲线优化算法，在对地铁列车的运行过程进行受力分析以及对速度曲线的优化框架进行设计的基础上，应用 DQN 的强化学习架构实现了速度

表 4 不同线路和车型下的牵引能耗对比

测试案例	算法	牵引能耗/kWh	节能比例/%
线路 1/ A 型车	传统算法	34.350	6.95
	本文算法	31.964	
线路 1/ B 型车	传统算法	27.394	7.39
	本文算法	25.370	
线路 2/ A 型车	传统算法	26.788	5.45
	本文算法	25.328	
线路 2/ B 型车	传统算法	22.817	6.71
	本文算法	21.286	

曲线的寻优求解。

通过在 Matlab 平台上搭建仿真环境，并对传统算法和本文算法进行多场景仿真测试，验证本文算法的有效性。仿真结果显示，相比传统算法，本文算法实现了约 5%~7% 的牵引节能，表明本文算法可以实现节能效果的提升，具备一定的应用价值。

真实运营环境相比仿真环境更加复杂多变，为适应更具挑战性的实际需要，本文算法需与现有的 ATO 系统深度融合。对于 ATO 系统严苛的实时性要求，可采用“离线优化+在线调整”的二级策略：先在离线状态下训练 DQN 模型，生成一系列针对不同时段、不同载客量的基准节能速度曲线或 Q 网络，将其预置于车载控制器中；在列车实际运行时，系统再根据当前的初始条件选择或计算最优的速度曲线。该策略可有效规避 DQN 算法训练过程算力要求高、耗时长弊端，充分发挥 Q 网络在部署后计算速度快、稳定性高的优势，满足信号系统的运行要求。

此外，面对诸如临时限速、信号故障、站台门故障导致的额外停车以及乘客数量变化引起的列车质量动态改变等随机干扰，可通过将实时采集的列车状态信息（如当前位置、速度、能耗数据）和前方信号信息作为列车模型的动态输入，使算法能够对突发事件做出快速响应，实时微调速度曲线，增强算法的鲁棒性和自适应性，确保整个节能方案能安全稳定运行。

综上所述，本文提出的基于 DQN 的速度曲线优化方法在仿真层面验证了其节能效果与应用价值。同时，可通过设计“离线优化+在线调整”的部署策略以及结合实时信息反馈进行动态调整，在真实、复杂的运营环境中落地应用。

### 参考文献：

[1] 中国城市轨道交通协会. 城市轨道交通 2020 年度统计和分析报告 [R]. 北京: 中国城市轨道交通协会, 2021.

[2] 中国城市轨道交通协会. 城市轨道交通 2021 年度统计和分析报告 [R]. 北京: 中国城市轨道交通协会, 2022.

(下转第 265 页)