

基于Q-learning的高速铁路列车动态调度方法

韩忻辰¹, 俞胜平^{1†}, 袁志明², 程丽娟¹

(1. 东北大学 流程工业综合自动化国家重点实验室, 辽宁 沈阳 110819;

2. 中国铁道科学研究院集团有限公司 通信信号研究所, 北京 100081)

摘要: 高速铁路作为国家综合交通运输体系的骨干核心, 近十年来取得了飞速蓬勃的发展. 其飞速发展的同时也引发了路网复杂化、分布区域广等现象, 这些现象对高铁动态调度提出了更高的要求. 突发事件的不确定性会对列车造成时间延误影响, 甚至时间延误会沿路网传播, 造成大面积列车到发晚点. 而目前对于此问题的人工调度方式, 前瞻性及针对性较差, 难以对受影响列车进行迅速调整. 针对上述问题, 本文建立了以各列车在各车站延误时间总和最小为目标函数的高速铁路列车动态调度模型, 在此基础上设计了用于与智能体交互的仿真环境, 采用了强化学习中的Q-learning算法对模型进行求解. 最后通过仿真实例验证了仿真环境的合理性以及Q-learning算法用于高铁动态调度的有效性, 为高铁调度员做出优化决策提供了良好的依据.

关键词: 高速铁路列车; 动态调度; 强化学习; Q-learning

引用格式: 韩忻辰, 俞胜平, 袁志明, 等. 基于Q-learning的高速铁路列车动态调度方法. 控制理论与应用, 2021, 38(10): 1511 – 1521

DOI: 10.7641/CTA.2021.00612

High-speed railway dynamic scheduling based on Q-learning method

HAN Xin-chen¹, YU Sheng-ping^{1†}, YUAN Zhi-ming², CHENG Li-juan¹

(1. State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang Liaoning 110819, China;

2. Signal & Communication Research Institute, China Academy of Railway Sciences Corporation Limited, Beijing 100081, China)

Abstract: As the backbone of the national comprehensive transportation system, high-speed railway has achieved rapid and vigorous development in the past decade. At the same time, its rapid development has also caused the phenomena of complicated road networks and wide distribution areas. These phenomena have placed higher requirements on high-speed railway scheduling. Unexpected events will affect the time delay of trains, and even the delay time will spread along the road network, causing large-area trains to arrive or departure late. However, the manual scheduling method is poorly forward-looking and pertinent, and it is difficult to quickly adjust the affected trains. In view of the above problems, this paper establishes a high-speed railway dynamic scheduling model with the minimum the sum of delay time as the objective function. Based on this model, an environment for interacting with the agent is designed, and the Q-learning algorithm is used to solve the model. Finally, the simulation examples verify the rationality of the simulation environment and the effectiveness of the Q-learning algorithm for the dynamic scheduling problem. It can provide a good basis for dispatchers to make more optimal decisions.

Key words: high-speed railway; dynamic scheduling; reinforcement learning; Q-learning

Citation: HAN Xincheng, YU Shengping, YUAN Zhiming, et al. High-speed railway dynamic scheduling based on Q-learning method. *Control Theory & Applications*, 2021, 38(10): 1511 – 1521

1 引言

随着我国高速铁路列车事业的飞速发展, 现我国高铁运行里程数已达三万余公里, 居世界第一. 也正是随着高铁速度的不断提升与列车数量的不断增多,

使得高铁路网规模扩大化, 路网结构复杂化, 其强耦合、快演变、多约束等特点也越来越明显. 加之, 恶劣天气、设备故障等突发事件的不可预知性, 可能会导致单列车或多列车出现延误晚点现象. 随着路网不断

收稿日期: 2020-09-10; 录用日期: 2021-03-25.

[†]通信作者. E-mail: spyu@mail.neu.edu.cn; Tel.: +86 13700033130.

本文责任编辑: 阳春华.

国家自然科学基金项目(U1834211, 61790574, 61603262, 61773269), 辽宁省自然科学基金(2020-MS-093)资助.

Supported by the National Natural Science Foundation of China (U1834211, 61790574, 61603262, 61773269) and the Natural Science Foundation of Liaoning Province (2020-MS-093).

传播扩散,将造成大面积的列车无法按照计划运行。因此,当原调度计划受到扰动影响无法继续运行时,如何进行重调度或动态调度^[1],如何能更好地协调各列车尽快恢复有序运行、减小延误时间、缩小影响范围,从而实现高速铁路列车运行快速、运营安全、到站准时、旅客舒适,已成为目前迫切需要解决的问题^[2]。

目前,列车调度方法研究主要集中在静态调度即运行图编制上,而针对高速铁路列车动态调度问题的研究,其方法大致可分为仿真方法、运筹学方法和智能算法3类。利用仿真方法有文献[3]针对随机扰动交通条件下最优调度方案的稳定性,提出了一种将基于替代图的铁路优化系统(railway optimization by means of alternative graphs, ROMA)与路网设计微观仿真环境(environment for the design and simulation of railway network, EGTRAIN)相结合的框架。文献[4]提出了一种基于仿真方法的列车时刻表优化模型,并在该模型的基础上又开发了一种集成路径再规划算法的框架。文献[5]则基于全路路网,利用MapInfo技术整合铁路车站及区间基础地理信息数据,构建了一个列车运行仿真系统。文献[6]基于部件组合思想和考虑线路双向运行以及列车运行过程中随机扰动建立的列车运行仿真模型,可根据需要模拟的列车运行计划及关联路网的数据,快速生成与列车运行环境和列车运行规则对应的仿真框架,并且在最终的仿真验证中也证实了该模型的有效性。而基于运筹学方法的如文献[7-9]结合累积流变量模型的特点,构建基于累积流变量的列车运行图整数规划模型,将原问题按列车划分为子问题并通过拉格朗日松弛算法优化列车运行图。文献[10-11]都是针对不同扰动进行的研究,文献[10]主要是针对若干典型场景建立了不同模型并采用改进粒子群算法对模型求解,而文献[11]则是通过建立不同策略匹配表,并基于极大加代数的时刻表递推思路进行求解。文献[12]提出了针对新增运行线(包括将需调整的运行线)视为一类特殊的新增列车的整数规划问题,并建立了基于拉格朗日松弛法的局部调整模型。运用智能算法的如文献[13]针对突发事件的影响,提出了调整策略控制参数的粒子群优化算法,并同常规的遗传算法(genetic algorithm, GA)与粒子群优化(particle swarm optimization, PSO)算法进行了比较。文献[14]采用了蚁群优化算法与遗传算法,其最终结果显示该两种算法的延迟成本比先进先出(first input first output, FIFO)的方法降低了30%和28%。文献[15]采用双适应值法,通过改变3个不同的参数分析粒子群算法所得结果,可以有效处理大规模列车约束问题。文献[16]将人工免疫算法引入到遗传算法中,指导个体向最优方向发展,能够避免一般进化算法可能遇到的退化现象,提高全局搜索能力。文献[17]通过改进萤火虫算法(improved firefly algorithm, IFA)优化调

整列车调度,在保证列车安全运行的情况下,可以将列车晚点产生的不利影响控制在很小的区间范围内。

仿真方法所构建的模型较为直观,但是其基于一定的假设条件之上,真正遇到大规模问题就会显得乏力,模型的结果与现实情况会有很大的差距。运筹学方法缺乏实时性和适应性,不能很好地满足实际列车动态调整的需要^[13]。智能算法容易陷入局部最优,且在多约束条件下,为求得更好更优的解需要大量增加迭代次数,导致计算时间过长。而强化学习是以目标为导向,通过智能体与环境不断交互学习进而求得最优策略的算法,采用强化学习求解组合优化问题,通过奖励函数的合理引导可以取得比人类设计更好的解决方法,且模型的泛化能力强。并且目前采用该算法应用于高速铁路列车动态调度问题的研究还是相对较少。

因此,本文针对突发事件对高速铁路列车造成的延误影响,结合强化学习中智能体与环境不断交互、不断试错的特点,选定具体的单线单向多车站且列车可在多股道车站内进行越行的场景,采用经典的Q-learning方法进行调度策略的优化。首先建立高速铁路列车动态调度的数学模型,然后根据模型搭建起用于与智能体交互的环境,运用Q-learning方法使智能体在不同状态下或探索未知的策略,或利用当前的最优策略,循环往复不断优化,最终得到调整后的调度计划,将延误影响降至最小。将所建立的模型与算法在多种扰动场景下进行仿真研究,仿真结果表明应用强化学习中的Q-learning方法能够学习到满意的高铁调度策略,为高铁优化调度提供良好的决策依据。

2 高速铁路列车动态调度问题描述

在高铁列车实际运行过程中,当遇到大风、暴雪等突发事件影响导致列车到发时间晚点时,需要根据初始调度计划信息、区间运行时间和车站停站等信息,在初始调度计划的基础上,对列车在所停车站的到发时间、列车顺序、股道占用等进行快速调整,使受影响列车尽可能地恢复至原计划运行状态,避免因一车延误造成全网延误的现象。因此高速铁路列车动态调度问题是在初始调度计划安排的基础上,根据扰动的影响程度,对其进行再次的决策与优化。而再次的决策与优化是在某一评价函数最优(如总晚点时间最小或总晚点列车数量最少)的情况下,对突发扰动影响的列车及其相关联的列车确定闭塞区间的占用时间、到发时间及列车顺序等一系列事件的动态调度。

现有的依靠调度员经验按列车逐站调整或者整体平移的人工调整方式,难以在突发事件下对列车运行图进行快速准确地调整,往往只是将延误列车及其后续列车进行顺延处理,并不能将延误所带来的影响降至最小。所以本文采用Q-learning的方法,将延误影响

到的列车群当作一个整体, 进行包括数学模型、交互环境、状态动作、奖励函数等关键因素的设计, 从而对其进行优化调度。

3 高速铁路列车动态调度数学模型

3.1 符号定义

高速铁路列车动态调度数学模型中所用模型参数定义如表1所示。

表 1 模型参数定义

Table 1 Model parameter meaning table

符号参数	参数含义
i	列车序号, $i = 1, 2, 3, \dots, N$, N 表示列车总数量。
j	车站序号, $j = 1, 2, 3, \dots, M$, M 表示车站总数量。
i^*	发生离站晚点的列车。
j^*	发生离站晚点的车站。
$T_{i^*j^*}^*$	发生离站晚点的 i^* 车在 j^* 站的延误时间。
$k_{j,j+1}$	$k_{j,j+1} = 1, 2, 3, \dots, K_{j,j+1}$, $K_{j,j+1}$ 表示 j 至 $j+1$ 站之间闭塞区间的总数量。
l_j	j 站内股道的序号, $l_j = 1, 2, 3, \dots, L_j$, L_j 表示 j 站内股道总数量。
$z_{i_1,i_2,1}^0$	图定计划中, i_1 车与 i_2 车 在始发站的离站顺序参数。
A_{ij}^0	i 车在 j 站的图定计划到站时间。
D_{ij}^0	i 车在 j 站的图定计划离站时间。
$D_{i^*j^*}^*$	受扰动影响的 i^* 车在 j^* 站的离站时间。
T_j^a	在车站 j , 相邻列车之间最小的 到站时间间隔。
T_j^d	在车站 j , 相邻列车之间最小的 离站时间间隔。
$T_{ij}^{\min}$	i 车在 j 站的最小停站时间。
$T_{i,j,j+1}^{\min}$	i 车在 $j, j+1$ 站之间的最小区间运行时间。
$B_{k_{j,j+1}}$	$k_{j,j+1}$ 闭塞区间内列车的数量。
S_{l_j}	l_j 股道内列车的数量。
决策变量	变量含义
A_{ij}	受扰动影响, i 车到达 j 站的实际到站时间。
D_{ij}	受扰动影响, i 车离开 j 站的实际离站时间。
$z_{i_1,i_2,j}$	受扰动影响, i_1 车与 i_2 车 在 j 站的实际离站顺序参数。

3.2 数学模型

根据第2节高速铁路列车动态调度问题描述, 建立如下数学模型。

目标函数为最小化各列车在各车站的实际到站、离站时间与计划到站、离站时间的偏差和。

目标函数:

$$\text{Min} \sum_{i=1}^N \sum_{j=1}^M [(A_{ij} - A_{ij}^0) + (D_{ij} - D_{ij}^0)]. \quad (1)$$

约束条件:

1) 列车 i 实际离开车站 j 的时间不得早于图定离站时间. 而对于在 j^* 站受扰动影响发生晚点的 i^* 车而言, 其离站时间 $D_{i^*j^*}^*$ 为图定计划时间 $D_{i^*j^*}^0$ 与扰动时间 $T_{i^*j^*}^*$ 之和, 故离站时间约束如下所示:

$$\begin{cases} D_{ij} \geq D_{ij}^0, \\ D_{i^*j^*}^* = D_{i^*j^*}^0 + T_{i^*j^*}^*. \end{cases} \quad (2)$$

2) 列车 i 在车站 j 的停站时间不得小于 j 站规定的最小停站时间约束. 即 i 车在 j 站的离站时间 D_{ij} 距离其到站时间 A_{ij} 的间隔应大于等于该站最小的停站时间 $T_{ij}^{\min}$.

$$D_{ij} - A_{ij} \geq T_{ij}^{\min}. \quad (3)$$

3) 实际离站顺序参数 $z_{i_1,i_2,j}$ 约束. 因本文所研究场景为列车只可在多股道车站内进行越行, 不允许在线路上越行, 所以在前一站列车的离站顺序, 即为下一站的到站顺序, 故无须再对到站顺序进行约束判断。

$$z_{i_1,i_2,j} = \begin{cases} 0, & D_{i_1,j} > D_{i_2,j}, \\ 1, & D_{i_1,j} < D_{i_2,j}. \end{cases} \quad (4)$$

4) 相邻列车 $i, i+1$ 在车站 j 的最小进站时间间隔约束. 当 $z_{i,i+1,j-1} = 1$ 时, 即 i 车在 $i+1$ 车之前到达 j 站, 所以 $A_{i+1,j} - A_{ij}$ 应大于等于 j 站的最小到站间隔 T_j^a , 反之亦然。

$$\begin{aligned} (A_{ij} - A_{i+1,j})(1 - z_{i,i+1,j-1}) + \\ (A_{i+1,j} - A_{ij}) \cdot z_{i,i+1,j-1} \geq T_j^a, \end{aligned} \quad (5)$$

其中 $j \geq 2$ 。

5) 相邻列车 $i, i+1$ 在车站 j 的最小离站时间间隔约束. 当 $z_{i,i+1,j} = 1$ 时, 即 i 车在 $i+1$ 车之前离开 j 站, 所以 $D_{i+1,j} - D_{ij}$ 应大于等于 j 站的最小离站间隔 T_j^d , 反之亦然。

$$\begin{aligned} (D_{ij} - D_{i+1,j})(1 - z_{i,i+1,j}) + \\ (D_{i+1,j} - D_{ij}) \cdot z_{i,i+1,j} \geq T_j^d, \end{aligned} \quad (6)$$

其中 $j < M$ 。

6) 列车 i 在车站 $j, j+1$ 之间的最小区间运行时分约束. 受地形制约等因素的影响, 列车 i 在车站 $j, j+1$ 之间的区段内运行速度须保证列车安全, 即列车在 $j+1$ 站的到站时间 $A_{i,j+1}$ 与在 j 站的离站时间 D_{ij} 之差应大于等于该区间内的最小运行时分 $T_{i,j,j+1}^{\min}$ 。

$$A_{i,j+1} - D_{ij} \geq T_{i,j,j+1}^{\min}. \quad (7)$$

7) 当 $z_{i_1,i_2,j} = 1$ 时, 列车 i_1, i_2 在车站 j 的越行约束. 首先该车站内的股道数 L_j 应大于等于2; 其次发生越行的列车之间也必须满足进站、离站的时间间隔约束。

$$\begin{cases} L_j \geq 2, \\ D_{i_2,j} - D_{i_1,j} \geq T_j^d, \\ A_{i_1,j} - A_{i_2,j} \geq T_j^a. \end{cases} \quad (8)$$

8) j 与 $j+1$ 站之间的闭塞区间资源占用唯一性约束. 在 $k_{j,j+1}$ 闭塞区间内, 列车数量只有两种状态, 即无列车状态0或只有一辆列车状态1.

$$B_{k_{j,j+1}} \in \{0, 1\}. \quad (9)$$

9) j 站内股道资源占用唯一性约束. 同约束(8), 在 l_j 股道内, 列车数量也只有两种状态, 即无列车状态0或只有一辆列车状态1. 另外, 因受车站站内股道总数量限制, 停在该站列车 $\sum_{l_j=1}^{L_j} S_{l_j}$ 应小于等于该站内的总股道数 L_j .

$$\begin{cases} S_{l_j} \in \{0, 1\}, \\ \sum_{l_j=1}^{L_j} S_{l_j} \leq L_j. \end{cases} \quad (10)$$

上述建立的数学模型为混合整数非线性规划模型(mix-integer non-linear programming, MINLP), 其包含着大量的约束方程、离散变量和连续变量, 采用运筹学方法和智能优化方法均难以在有效时间内进行快速求解, 无法满足现场对实时性的要求^[18]. 而强化学习是以目标为导向的机器学习方法, 设计合理的交互环境, 其可以在模型未知(model-free)的情况下, 进行学习求解, 故采用强化学习中的Q-learning算法对其进行动态调度.

4 Q-learning算法实现

机器学习可根据反馈的不同, 将其分为监督学习(supervised learning)、无监督学习(unsupervised learning)和强化学习(reinforcement learning)3大类. 其中强化学习是一种以环境反馈作为输入的、特殊的、适应环境的机器学习方法, 它的主要思想是与环境交互和试错, 利用评价性的反馈信号实现决策的优化^[19], 其基本框架如图1所示.

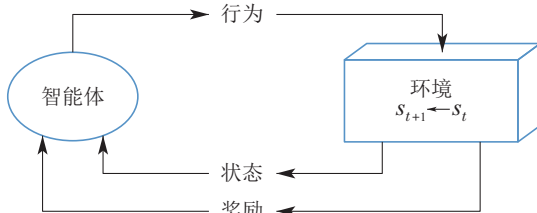


图1 强化学习框架

Fig. 1 Framework of RL

在强化学习中, Q-learning是目前应用最为广泛的算法之一, 其被公认为是强化学习发展过程中的一个里程碑^[20], 该算法的思想是直接优化一个可迭代计算的Q函数. 其中Q函数定义为在某一状态 $s \in S$ 时, 执

行动作 $a \in A$, 此后按最优动作序列执行所获得的回报的期望. Q函数更新公式为

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} - Q(s_t, a_t) + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1})], \quad (11)$$

其中: α 为算法收敛的学习率, γ 为折扣因子, r_{t+1} 为智能体在状态 s_t 采取行动 a_t 后所获得的即时奖励^[21].

Q-learning算法的运用, 还需定义其6个基本的组成要素, 分别为智能体(agent)及其可以采取的动作(action)、智能体进行交互的环境(environment)、智能体与环境交互后确定当前所处的状态(state)、采取不同的策略所能获得的奖励(reward)以及保存Q函数值的Q-table.

4.1 环境建立

根据第3.2节所设计的数学模型, 建立智能体与交互的环境. 建立交互环境的目的是让智能体不断试错寻优, 用以学习到最优策略. 所以环境设立的好坏也影响了最终决策的优劣. 在建立环境之前, 首先提出4个为方便处理问题的假设条件.

假设1 假设在环境中运行的列车为一个质点, 即忽略列车的长度. 这样处理的好处是会忽略列车在跨越两个闭塞区间资源或驶入、驶离车站时因车身长度所耗费的时间.

假设2 假设时间是离散的, 并且确定时间刻度的分辨率为1 min, 即智能体从环境获得观测值(observation)并采取动作是每分钟进行一次. 该假设是与人工调度以分钟为最小调度单位相符合的.

假设3 假设车站站内股道的设置是相对简单的, 仅仅是用一系列平行的直线来代替, 而不去考虑现实场景股道之间的复杂性.

假设4 假设每辆列车只有两个动作前进(go)或者停止(stop), 如果列车选择前进动作, 那么列车将按规定速度前进, 忽略列车提速的过程. 同样, 也忽略列车减速过程.

由上述提出的4个假设, 结合数学模型中的约束条件, 建立起实验所需的环境. 首先对列车可占用的资源进行编号, 其中可占用的资源包括: 车站内的股道以及车站间的闭塞区间. 编号顺序以简单的两站一区间为例进行说明, 如下图2所示.

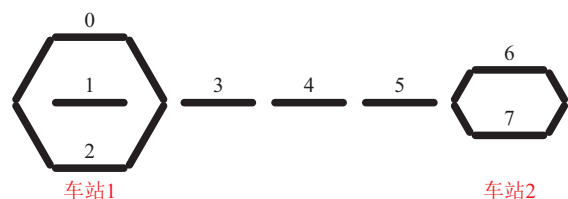


图2 两站一区间资源编号

Fig. 2 Resource number of two stations and one section

上图所示为车站1内有3个股道, 编号为0, 1, 2. 车站1, 2之间的整个区间被分为3个闭塞区间, 编号为3, 4, 5. 车站2内有两个股道, 编号为6, 7.

下面考虑数学模型中所需要满足的约束条件.

$$S = [s_{i,t}]_{N \times T} = \begin{bmatrix} s_{1,1} & \cdots & s_{1,t} & \cdots & s_{1,T} \\ \vdots & & \vdots & & \vdots \\ s_{i,1} & \cdots & s_{i,t} & \cdots & s_{i,T} \\ \vdots & & \vdots & & \vdots \\ s_{N,1} & \cdots & s_{N,t} & \cdots & s_{N,T} \end{bmatrix}. \quad (12)$$

针对约束(1)–(7), 本文建立了一个列车状态矩阵 S , 其具体的表现形式如式(12). 其中: S 中的第 i 行表示第 i 辆列车, 第 t 列表示时刻 t , 列的增加表示时刻 t 的增加. S 中的每个元素 $s_{i,t}$ 表示在第 t 时刻, 第 i 辆车所占据的资源编号. N 表示需要进行动态调度的列车总数, T 表示经动态调度后所有列车到达终点站的时间总跨度.

例如式(13), S 中第2列, 表示在第1分钟(时刻 t 从0时刻开始), G1车在0号资源, G2车在1号资源, G3车在2号资源. 假设图3情景发生在第2分钟, 此时G1车选择前进, 对应 S 中第3列各车所对应的资源占用. 由此可见, S 中保留了各列车占用资源的历史信息, 所以时间约束(1)–(7)可以从状态矩阵 S 中进行判断.

$$S = \begin{bmatrix} 0 & 0 & 3 & 3 & 4 & 5 & \cdots \\ 1 & 1 & 1 & 1 & 3 & 3 & \cdots \\ 2 & 2 & 2 & 2 & 2 & 2 & \cdots \end{bmatrix}. \quad (13)$$

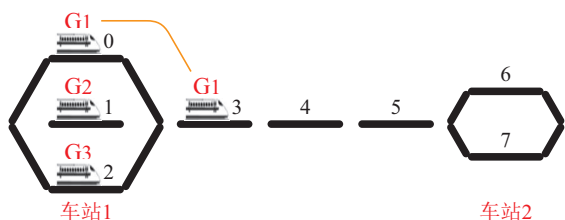


图3 状态切换

Fig. 3 State switching

约束(8)–(9)都可以归为资源占用的唯一性约束, 即每个资源在当前时刻只可为空闲或最多被一辆列车所占据, 不可出现多列车占用资源冲突的情况, 如图4所示. 如果在某一时刻出现列车争夺资源的现象, 就把该资源随机分配给其中一辆车.

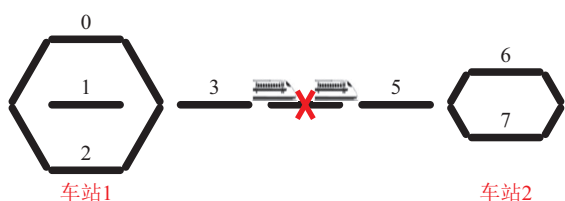


图4 资源唯一性约束

Fig. 4 Uniqueness of resources

综上所述, 模型中的约束条件均在建立的环境中一一体现, 并限制了智能体的动作选择.

4.2 智能体及其状态

前面提到了目前高速铁路列车动态调度的缺点, 本文所采取的方法是将整个因突发事件影响受到牵连的所有列车当作列车群, 作为一个智能体去进行调度策略的优化.

而对于智能体所处的状态定义, 则是根据第4.1节所建立的仿真环境, 将智能体的状态定义为每辆列车目前所处的资源编号. 如 $state = [i, j, k]$ 表示智能体包含3辆列车, 其目前所占用的资源分别为 i, j, k 号资源.

4.3 动作集合的定义及 ϵ -greedy 策略

对于每辆列车而言, 在当前时刻 t 可选择动作分为两种, 即前进(go)或停止(stop). 而对于第4.2节所定义的智能体, 其全部可以采取的动作组合数最多为 2^N 种, 其中 N 表示智能体所包含的列车数量.

当列车处于闭塞区间资源并满足第3.2节数学模型中的所有约束时, 其可以采取的动作集合为 $\{go, stop\}$, 为加快算法的收敛, 本文将对该列车强制采取动作前进(go). 又为了使算法可以探索到更多的可能性, 也为了发现可能存在的更优策略, 对处在车站资源的列车且满足模型中所有约束时, 将从动作集合 $\{go, stop\}$ 中依策略进行动作选择, 而并非是直接采取动作前进(go). 以图5来说明上述提到的两种情况.

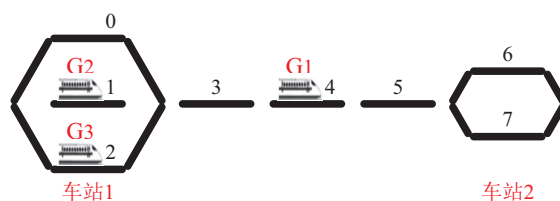


图5 采取动作时的两种情况

Fig. 5 Two situations while taking an action

目前, G1车正处于4号资源, 在该资源中必须连续前进, 选择动作前进. 若满足所有约束, 则前进到5号资源. G2车目前正处于资源1, 当其满足所有约束时, 其可选的动作集合为 $\{go, stop\}$, 而并非强制采取前进. 这样处理的原因是本文所建立的模型仅允许列车在多股道车站内进行越行, 所以可能G2车停在该站让G3车先去占用3号资源会得到一个更好的解.

其次, 本文利用 ϵ -greedy 策略去进行动作的选择. 采用这种策略的好处是在保证智能体往更优方向前进的同时又可以保留一定的探索性. 其中

$$\epsilon = \frac{0.5}{1 + e^{\frac{10 \times (\text{episode} - 0.6 \times \text{max_episode})}{\text{max_episode}}}}. \quad (14)$$

ϵ -greedy 策略如下:

$$\mu(a_t | s_t) = \begin{cases} \text{random } A(s_t), & \text{rand} > 1 - \epsilon, \\ a^*, & \text{其他.} \end{cases} \quad (15)$$

式(15)中: a^* 表示在 s_t 状态下 Q 值最大的动作, $A(s_t)$ 表示在 s_t 状态下所有可选的动作集合, rand 为一个服从标准正态分布的抽样值. 根据式(14)绘制出 ϵ 与 $1 - \epsilon$ 的图像如图6所示.

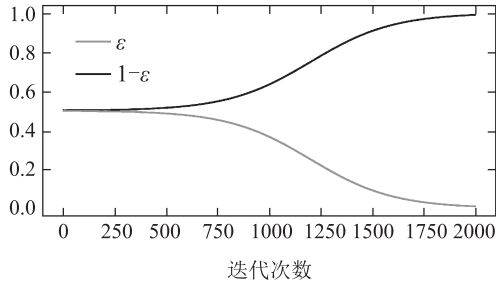


图6 ϵ 与 $1 - \epsilon$ 的图像

Fig. 6 ϵ and $1 - \epsilon$

可发现随着迭代次数的增长, ϵ 逐渐减小趋近于0, 而 $1 - \epsilon$ 则逐渐增大趋近于1. 表示在训练的开始, 智能体大概有50%的概率去探索新的动作. 但是随着训练次数的增加, 智能体更偏向于利用已经学习到的知识, 从而会更多地采用已经学习到的最优动作序列.

4.4 奖励函数及奖励机制

因为在第3.2节中数学模型的优化目标为

$$\min \sum_{i=1}^N \sum_{j=1}^M [(A_{ij} - A_{ij}^0) + (D_{ij} - D_{ij}^0)],$$

即最小化所有列车在各站的到发晚点时间和. 所以本文所采取的奖励函数为目标函数值的倒数, 即

$$\Omega = \frac{1}{\sum_{i=1}^N \sum_{j=1}^M [(A_{ij} - A_{ij}^0) + (D_{ij} - D_{ij}^0)]}. \quad (16)$$

根据上式发现, 当总延迟时间越小, 即目标函数越优, 给予智能体的奖励也就越大. 但是注意奖励机制的设置是只有当智能体到达最终状态时才会给予奖励, 即所有列车都到达终点站时才会有奖励, 其余时刻智能体所获得奖励为0. 在这种奖励机制与奖励函数的设置下, 只有在完整的一轮训练结束后, 才会反向传播更新Q-table的值. 这样设置的原因是根据本文数学模型中所设定的目标函数及奖励函数, 只有当所有列车均到站后才可以确定, 类似问题的处理方法在文献[22]也有所涉及.

在实验中发现当训练次数增多, 该奖励反向传播至前几轮经过的状态时, 其对应Q-table中的值非常小, 故在式(16)基础上, 对其做了放大10000倍的处理.

另外, 本文在奖励函数的设置中还采用了一种加速算法收敛的技巧, 即如果智能体表现的不好, 并非置之不顾, 给予奖励0, 而是会给予其一惩罚. 当扰动发生造成的延误时间已知时, 对于整个智能体而言其完成一轮训练的时间阈值可人为确定. 当目标函数值超过该阈值时, 即该解已经不满足调度要求的下限,

认为智能体在该轮训练中表现较差, 给予的奖励为一个负值. 综上所述, 奖励 r 的表达式为

$$r = \begin{cases} 10000\Omega, & \text{object} \leq T_{\text{threshold}}, \\ -1, & \text{其他}. \end{cases} \quad (17)$$

4.5 Q-table的结构

首先, 本文是利用 t 与action去获得 $Q(s_t, a_t)$, 而不是常规地利用state与action去获得 $Q(s_t, a_t)$. 具体原因是Q-table的设置并没有采用常规state $\times$ action的结构, 因为在此算法中state的数量太多. 例如智能体包含列车数量为 N , 线路资源数量为 M , 时间跨度为 T 分钟, 那么该环境所对应state数量为 $M^N \cdot T$, 其对应的Q-table元素个数为 $M^N \cdot T \times 2^N$, 当列车数量或资源数量增加时, 该数值将呈现指数型的爆炸式增长.

故本文所建立的Q-table为time $\times$ action的结构, 如表2. 对于上述情况, Q-table中的元素个数仅为 $T \times 2^N$. 随着列车数量增加与时间跨度增大, 这样构造的Q-table也会扩张, 但是其扩张幅度远小于state $\times$ action类型的Q-table. 当然这样处理是通过牺牲一部分算法的计算时间来实现的, 即需要更多次的训练才可以得到理想的结果. 之后为方便理解与统一表达, 依旧采用 $Q(s_t, a_t)$ 这种表达方式.

表2 Q-table结构

Table 2 The structure of Q-table

time	action		
	a_1	$\dots$	a_A
t_1	$Q(s_{t_1}, a_1)$	$Q(s_{t_1}, \dots)$	$Q(s_{t_1}, a_A)$
t_2	$Q(s_{t_2}, a_1)$	$Q(s_{t_2}, \dots)$	$Q(s_{t_2}, a_A)$
$\dots$	$Q(\dots, a_1)$	$Q(\dots, \dots)$	$Q(\dots, a_A)$
t_T	$Q(s_{t_T}, a_1)$	$Q(s_{t_T}, \dots)$	$Q(s_{t_T}, a_A)$

4.6 Q-learning算法流程及伪代码实现

根据上述Q-learning算法所需的几大要素的设置, 给出Q-learning算法在求解高速铁路列车调度时的算法流程:

步骤1 定义超参数学习率 α 、折扣因子 γ , 导入列车时刻表T-Table, 同时确定扰动作用点delay_point, 造成的延误时间delay_time等. 初始化环境、智能体及其状态, 初始化Q-table, 列车状态矩阵S_matrix, 总训练回合数max_episodes.

步骤2 开始智能体的策略更新循环, 在每次训练之前初始化当前时刻 t , 初始化智能体的起始状态state.

步骤3 根据当前时刻 t 、状态state, 判断智能体当前时刻可采取的满足约束的动作集合actions, 以及下一时刻智能体中各列车可能到达的资源集合res.

步骤4 根据 t , actions与当前进行训练的回合

数episode以及Q-table运用 ϵ -greedy策略选择动作action.

步骤 5 根据 t , state以及action得到下一时刻的状态state₋, 奖励 r 以及是否可以停止本轮训练的标志done.

步骤 6 根据 $t + 1$, state₋获得智能体下一时刻满足约束的动作集合actions₋.

步骤 7 根据 t 以及action从Q-table中获得式(11)中 Q 函数更新公式中的 Q 预测值

$$Q_{\text{predict}} = Q(s_t, a_t). \quad (18)$$

步骤 8 根据Q-table与求得的actions₋去计算 $\max Q_- = \max_{a_{t+1}} Q(s_{t+1}, a_{t+1})$, 可以计算 Q 目标值

$$Q_{\text{target}} = r + \gamma \max Q_-. \quad (19)$$

步骤 9 至此式(11)中所有变量都已得到, 可进行 Q 函数的更新, 即对Q-table进行更新.

步骤 10 当前时刻 $t \leftarrow t + 1$ 更新, 状态state更新. 由done判断本轮循环是否可以结束, 若未结束则返回步骤3; 若本轮循环结束并且结果未收敛, 则返回步骤2, 并且 $\text{episode} \leftarrow \text{episode} + 1$; 否则进行步骤11.

步骤 11 如果智能体最后结果已经收敛或者已经达到最大训练回合数, 则退出训练并输出训练优化过后的调度计划opt_plan.

Q-learning算法求解高铁动态调度问题的伪代码如表3所示.

表 3 Q-learning求解高铁动态调度

Table 3 Q-learning solves the problem of high-speed railway dynamic scheduling

Input: $\alpha, \gamma, T\text{-Table}, Q\text{-table}, \max_episodes, S_matrix, \text{delay_point}, \text{delay_time}$	
Output: opt_plan	
1	init(environment, Q-table, episode);
2	while episode $\leq$ max_episodes do
3	init(t , state, done);
4	while !done do
5	action, res $\leftarrow$ Get_act_res(t , state);
6	state ₋ , r , done $\leftarrow$ Step(t , state, action, res);
7	max action ₋ $\leftarrow$ max($t + 1$, state ₋ , Q-table);
8	$Q(s_t, a_t) \leftarrow$
	$Q(s_t, a_t) + \alpha[r + \gamma \cdot \max\text{action}_- - Q(s_t, a_t)]$;
9	$t = t + 1$;
10	state = state ₋ ;
11	episode = episode + 1;
12	if Q-Table converge then
13	End While Loop;
14	opt_plan $\leftarrow$ greedy(Q-table);
15	plot(opt_plan);
16	End

5 仿真实验

5.1 环境参数

为验证所搭建仿真环境的合理性以及Q-learning算法解决高铁动态调度问题的有效性, 利用仿真实例进行验证. 该实例采用Python3.6编写, 运行环境为Windows10 64位操作系统, 处理器为Intel i7-8系列, 16 GB内存.

进行验证的例子采用三车四站三区间的情形, 其中21个资源构造如图7所示.

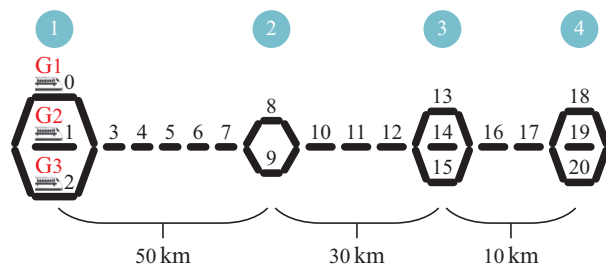


图 7 仿真环境设置

Fig. 7 Simulation environment

规定列车速度为300 km/h, 车站1到车站2距离50 km, 分为5个闭塞区间, 即通过每个闭塞区间需2 min; 车站2到车站3距离30 km, 分为3个闭塞区间, 即通过每个闭塞区间需2 min; 车站3到车站4距离10 km, 分为2个闭塞区间, 即通过每个闭塞区间需1 min. 规定第3.2节数学模型中的到发时间间隔约束 $T_j^a = T_j^d = 1$ min, 各区间最小运行时间为 $T_{i,1,2}^{\min} = 10$ min, $T_{i,2,3}^{\min} = 6$ min, $T_{i,3,4}^{\min} = 2$ min, 并规定经停站最小作业时间 $T_{ij}^{\min} = 3$ min.

列车图定计划时刻表如表4所示.

表 4 计划时刻表

Table 4 Table of scheduled times						
车站	1	2		3		4
车次	发	到	发	到	发	到
G1	11:03	11:13	11:16	11:22	11:25	11:27
G2	11:06	11:16	11:19	11:25	11:28	11:30
G3	11:10	11:20	11:23	11:29	11:32	11:34

突发情况设定为 $T_{G1,1}^* = 6$ min. 此时, 时间跨度 $T_{\text{span}} = 11:34 - 11:03 = 31$ min. 阈值可设定为 $T_{\text{threshold}} = T_{\text{span}} + T_{G1,1}^* + 10 = 47$ min, 即如果智能体不能在47 min内到达终点状态, 就给予其一惩罚.

5.2 仿真环境验证及算法收敛性分析

在第5.1节所示的仿真环境情形下, 对智能体进行大量训练, 最终经实验确定超参数 $\alpha = 0.1, \gamma = 0.9, \max_episodes = 2300$, 训练结果如图8所示.

在 $T_{G1,1}^* = 6$ min的情况下, 求得目标函数的最优解为42 min, 即当G1列车在始发站晚点6 min的情况下, 对3辆列车在4个站造成的总晚点时间为42 min, 调整后的调度计划如图9所示.

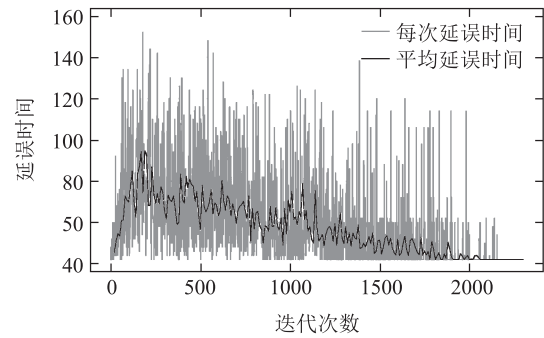


图 8 训练结果

Fig. 8 The results of training

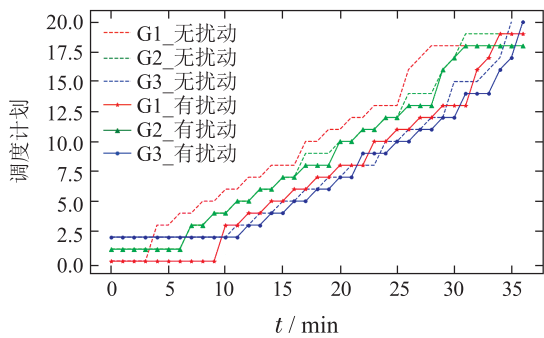


图 9 调整后的运行图

Fig. 9 The adjusted schedule

检查校验后,该调度计划满足数学模型中的所有约束,发现G2车在各站均按计划到站、离站;G1车因为扰动延误在G2发车后发车,在各站的到站、离站均晚点6 min,所以G1车的总晚点时间为36 min;G3车因避让晚点的G1车,在各站的到站、离站均晚点1 min,所以G3车的总晚点时间为6 min,最终目标函数的解为42 min.从图9中也可以发现G2,G3列车在站内调整计划占用的股道资源以满足资源占用唯一性约束.

图9对应调整后的时刻表如表5所示.

表 5 调整后的时刻表						
Table 5 The adjusted timetable						
车站	1	2		3		4
车次	发	到	发	到	发	到
G1	11:09	11:19	11:22	11:28	11:31	11:33
G2	11:06	11:16	11:19	11:25	11:28	11:30
G3	11:11	11:21	11:24	11:30	11:33	11:35

训练结束后,Q-table中各值(取整之后)如图10所示.

改变扰动延误时间及扰动作用点进行多种仿真.

1) 当 $T_{G1,2}^* = 2\text{ min}$, $T_{\text{threshold}} = 31 + 2 + 10 = 43\text{ min}$ 时,最终目标函数解为12 min,训练结果与动态调度计划如图11所示.

2) 当 $T_{G2,1}^* = 3\text{ min}$, $T_{\text{threshold}} = 31 + 3 + 10 = 44\text{ min}$ 时,最终目标函数解为24 min,训练结果与动

态调度计划如图12所示.

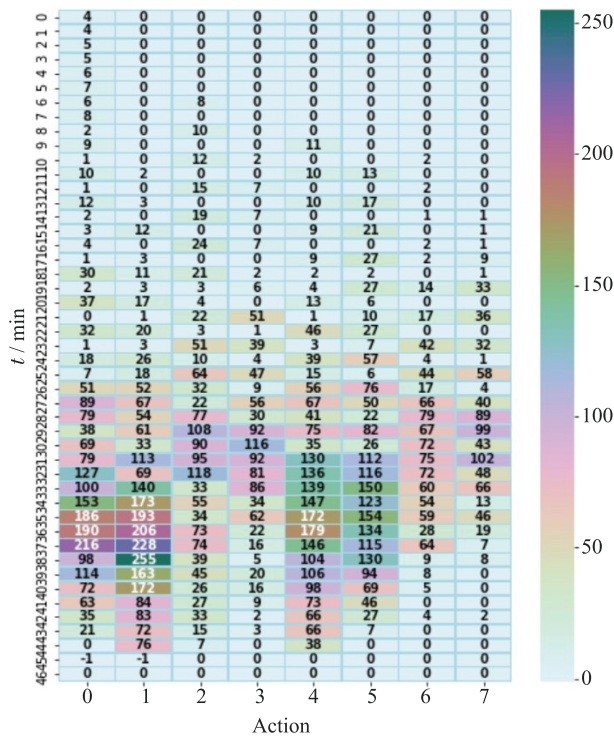


图 10 Q-table结果

Fig. 10 The Q-table

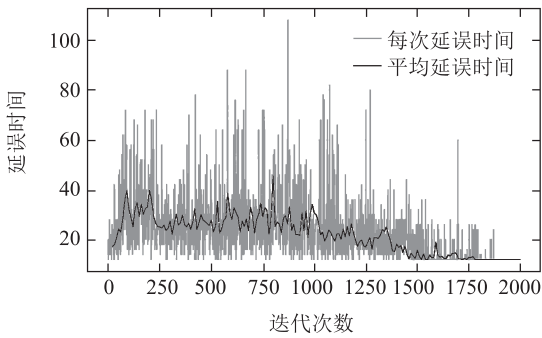


图 11 $T_{G1,2}^* = 2\text{ min}$ 仿真结果

Fig. 11 The simulation results of $T_{G1,2}^* = 2\text{ min}$

5.3 复杂场景验证及与人工调度方法对比

为更好地验证Q-learning算法在求解较大规模场景时的有效性,给出以下的验证场景如图13所示.同样,规定列车速度为300 km/h,根据图中数据计算得车站1,2之间每个闭塞区间的运行时间为2 min,车站

2, 3之间每个闭塞区间的运行时间为2 min, 车站3, 4之间每个闭塞区间的运行时间为1 min, 车站4, 5之间每个闭塞区间的运行时间为2 min, 车站5, 6之间每个闭塞区间的运行时间为2 min, 并且规定 $T_j^a = T_j^d =$

1 min, $T_{ij}^{\min} = 3$ min.

各列车计划时刻表如表6所示. 在该仿真场景下, 对不同的晚点时间、晚点车站及晚点列车做了多种仿真, 最终的调度结果如图14所示.

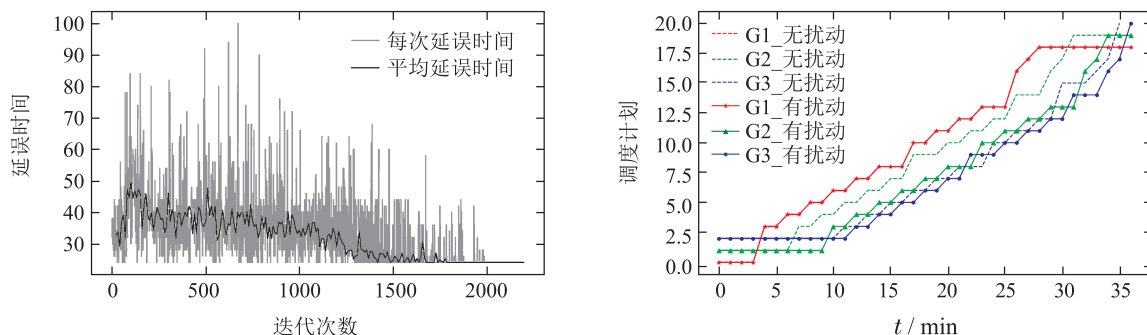


图 12 $T_{G2,1}^* = 3$ min 仿真结果

Fig. 12 The simulation results of $T_{G2,1}^* = 3$ min

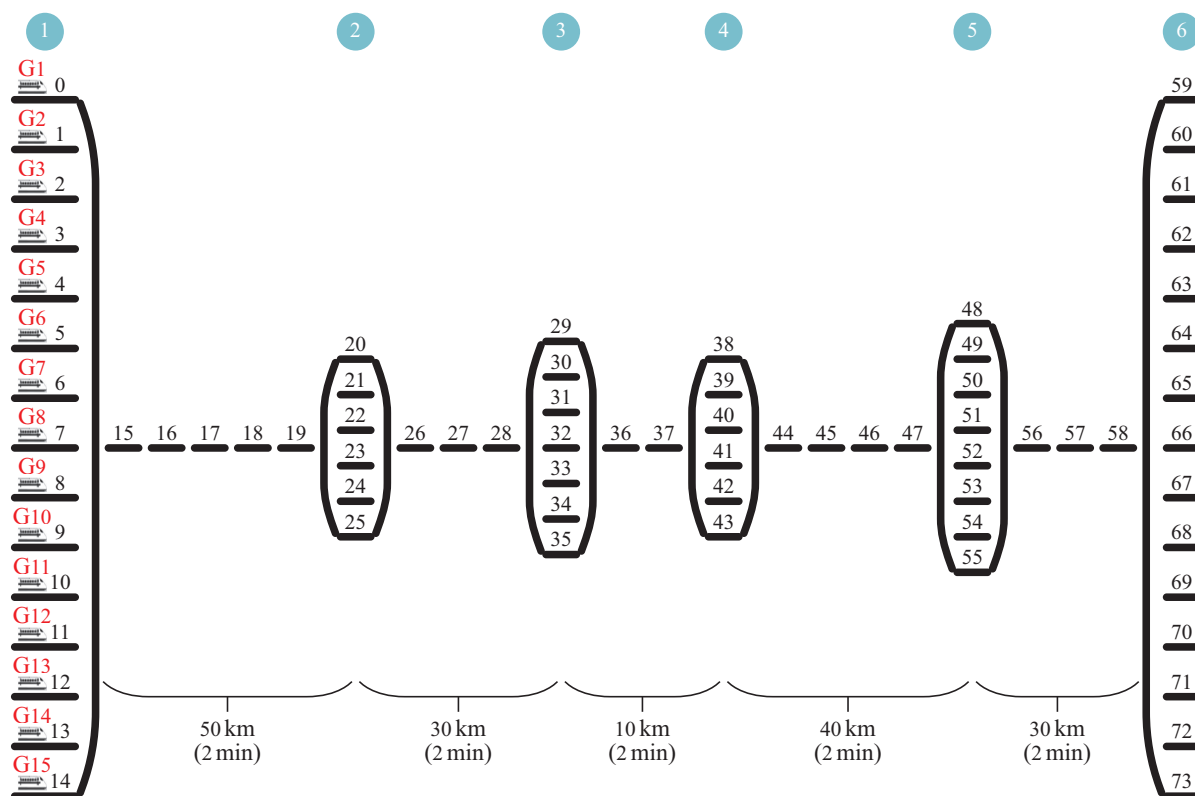
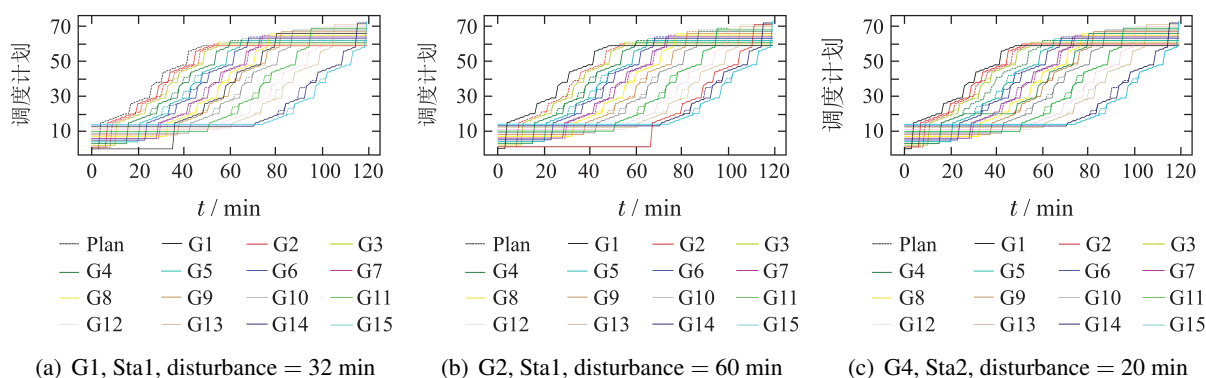


图 13 复杂场景实例

Fig. 13 The complex scenario



(a) G1, Sta1, disturbance = 32 min

(b) G2, Sta1, disturbance = 60 min

(c) G4, Sta2, disturbance = 20 min

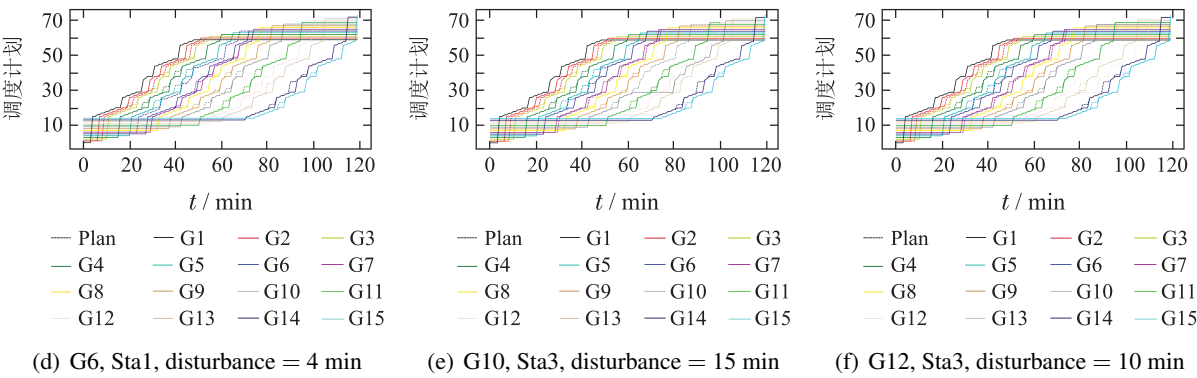


图 14 复杂场景仿真

Fig. 14 The complex scenario simulation

表 6 计划时刻表

Table 6 Table of scheduled times

车站	1	2		3		4		5		6
车次	发	到	发	到	发	到	发	到	发	到
G1	11:03	11:13	11:16	11:22	11:25	11:27	11:30	11:38	11:41	11:47
G2	11:06	11:16	11:19	11:25	11:28	11:30	11:33	11:41	11:44	11:50
G3	11:10	11:20	11:23	11:29	11:32	11:34	11:37	11:45	11:48	11:54
G4	11:15	11:25	11:28	11:34	11:37	11:39	11:42	11:50	11:53	11:59
G5	11:20	11:30	11:33	11:39	11:42	11:44	11:47	11:55	11:58	12:04
G6	11:23	11:33	11:36	11:42	11:45	11:47	11:50	11:58	12:01	12:07
G7	11:28	11:38	11:41	11:47	11:50	11:52	11:55	12:03	12:06	12:12
G8	11:32	11:42	11:45	11:51	11:54	11:56	11:59	12:07	12:10	12:16
G9	11:37	11:47	11:50	11:56	11:59	12:01	12:04	12:12	12:15	12:21
G10	11:42	11:52	11:55	12:01	12:04	12:06	12:09	12:17	12:20	12:26
G11	11:50	12:00	12:03	12:09	12:12	12:14	12:17	12:25	12:28	12:34
G12	11:55	12:05	12:08	12:14	12:17	12:19	12:22	12:30	12:33	12:39
G13	12:00	12:10	12:13	12:19	12:22	12:24	12:27	12:35	12:38	12:44
G14	12:10	12:20	12:23	12:29	12:32	12:34	12:37	12:45	12:48	12:54
G15	12:14	12:24	12:27	12:33	12:36	12:38	12:41	12:49	12:52	12:58

对于上述6种仿真情况, 分别将Q-learning方法与人工方法(Manual)的调度结果进行对比, 如表7所示. 其中, 人工方法采取的是晚点列车整体平移, 不产生越行的方式进行调整.

表 7 对比结果

Table 7 The comparison result

扰动情况	Q-learning/min	Manual/min	ψ /%
G1_Sta1_32	320	4800	93.3
G2_Sta1_60	600	8400	92.9
G4_Sta2_20	160	1920	91.7
G6_Sta1_4	50	400	87.5
G10_Sta3_15	90	540	83.3
G12_Sta3_10	60	240	75

其中, 改进率 ψ 的计算公式如式(20). 当 $\psi > 0$ 时, 说明调度结果通过Q-learning方法得到了改进; 反之, $\psi < 0$ 时, 说明Q-learning方法调度效果不如人工方

法.

$$\psi = \frac{\text{result(Manual)} - \text{result(Q)}}{\text{result(Manual)}} \times 100\%. \quad (20)$$

经过多种仿真实例, 可发现本文建立的数学模型及通过该模型所搭建的环境是满足约束、符合算法运行条件的. 在此基础上, 运用Q-learning算法对该模型进行了求解并同基本的人工调度方法进行了对比. 结果显示: 在合适的超参数、训练次数的条件下, 经过可接受的计算时间, 可以找到模型的满意解, 实现延误影响的最小化, 并对人工调度解有很大的提高.

6 结论

本文以高速铁路调度问题为研究对象, 建立了相应的数学模型, 在模型基础上又设计了合适的与智能体交互的仿真环境, 并利用强化学习中的经典Q-learning算法对不同情境下的实例进行了求解. 其最终结果表明, Q-learning算法要求的交互环境的设计是合

理的, Q-learning算法也适合求解高铁调度决策类型的问题。当然, 作为初步工作, 本文只是对单线单向线路进行了仿真, 还有许多其他复杂的因素需要考虑, 如线路成网、移动闭塞区间等较为复杂、细化的因素, 将在之后的工作中继续改进。

参考文献:

- [1] PANG Xinfu, GAO Liang, PAN Quanke, et al. Rescheduling methods of steelmaking-continuous casting with breakdown of converter or refining furnace. *Control and Decision*, 2015, 30(11): 1921 – 1929.
(庞新富, 高亮, 潘全科, 等. 某一转炉或精炼炉故障下炼钢-连铸生产重调度方法及应用. 控制与决策, 2015, 30(11): 1921 – 1929.)
- [2] ZHAO Hui, DAI Xuwu. Cooperative optimization method for high-speed trains running time and energy saving based on block sections. *Acta Automatica Sinica*, 2020, 46(3): 471 – 481.
(赵辉, 代学武. 基于闭塞区间的高速列车运行时间与节能协同优化方法. 自动化学报, 2020, 46(3): 471 – 481.)
- [3] QUAGLIETTA E, CORMAN F, GOVERDER M P. Impact of a stochastic and dynamic setting on the stability of railway dispatching solutions. *The 16th International IEEE Conference on Intelligent Transportation Systems*. Hague, Netherland: IEEE, 2013: 1035 – 1040.
- [4] HASANNAYEBI E, SAJEDINEJAD A, MARDANI S, et al. An integrated simulation model and evolutionary algorithm for train timetabling problem with considering train stops for praying. *Proceedings of the 2012 Winter Simulation Conference*. Berlin, Germany: IEEE, 2012: 1 – 13.
- [5] KONG Xu. *Research on train operation simulation system in railway station*. Chengdu: Southwest Jiaotong University, 2011.
(孔旭. 铁路车站列车运行仿真系统研究. 成都: 西南交通大学, 2011.)
- [6] YANG Rui, MOU Ruifang, YAN Haifeng. Research on simulation model of train operation plan. *China Railway Science*, 2017, 38(6): 125 – 133.
(杨锐, 牟瑞芳, 闫海峰. 列车运行计划仿真模型研究. 中国铁道科学, 2017, 38(6): 125 – 133.)
- [7] MENG L, ZHOU X. Simultaneous train rerouting and rescheduling on an N-track network: A model reformulation with network-based cumulative flow variables. *Transportation Research-Part B*, 2014, 67: 208 – 234.
- [8] LIAO Zhengwen, MIAO Jianrui, MENG Lingyun, et al. Optimization algorithm for train diagram of double-track railway based on Lagrange relaxation. *Journal of Railways*, 2016, 38(9): 1 – 8.
(廖正文, 苗建瑞, 孟令云, 等. 基于拉格朗日松弛的双线铁路列车运行图优化算法. 铁道学报, 2016, 38(9): 1 – 8.)
- [9] LI Gong. *coordinated optimization of train schedule, route and undercarriage dispatch adjustment*. Lanzhou: Lanzhou Jiaotong University, 2018.
(李功. 列车时刻表, 路径与车底调度调整的协同优化. 兰州: 兰州交通大学, 2018.)
- [10] LIU Min. A study on the methods for adjusting and optimizing high-speed railway train working diagram. *Railway Transport and Economy*, 2018, 40(10): 13 – 20.
(刘敏. 铁路客运专线列车运行图调整优化方法研究. 铁道运输与经济, 2018, 40(10): 13 – 20.)
- [11] LING Feixiang. *Research on method and evaluation of pattern train diagram on JingHu high-speed railway line*. Chengdu: Southwest Jiaotong University, 2018.
(凌飞翔. 京沪高速铁路规格化列车运行图编制理论与评价研究. 成都: 西南交通大学, 2018.)
- [12] JIANG Feng, NI Shaoquan, LV Hongxia. A high-speed railway new-added train timetable partial adjustment model based on lagrangian relaxation. *Journal of Transportation Systems Engineering and Information Technology*, 2018, 18(4): 163 – 170.
(江峰, 倪少权, 吕红霞. 基于拉格朗日松弛的高速铁路列车运行图新增运行线局部调整模型. 交通运输系统工程与信息, 2018, 18(4): 163 – 170.)
- [13] LIN Bo, YU Shengping, LIU Ziyuan, et al. High-speed railway dynamic scheduling method based on improved particle swarm algorithm. *Control Engineering of China*, 2021, 28(7): 1334 – 1341.
(林博, 俞胜平, 刘子源, 等. 基于改进粒子群算法的高铁动态调度方法. 控制工程, 2021, 28(7): 1334 – 1341.)
- [14] BO F, CLIVE R, PAUL W. A comparison of algorithms for minimizing delay costs in disturbed railway traffic scenarios. *Journal of Rail Transport Planning & Management*, 2012, 2(1/2): 23 – 33.
- [15] YU S P, LIN B, ZHANG T, et al. Dynamic scheduling method of high-speed trains based on improved particle swarm optimization. *2018 International Conference on Intelligent Rail Transportation*. Singapore: IEEE, 2018: 1 – 5.
- [16] ZHOU Xiaozhao. *Application of immune genetic algorithm in train operation adjustment of high-speed railway*. Beijing: Beijing Jiaotong University, 2014.
(周晓昭. 免疫遗传算法在高速铁路列车运行调整中的应用研究. 北京: 北京交通大学, 2014.)
- [17] XU Hu. A study on railway operation scheduling algorithm based on the improved firefly algorithm. *Microcomputer Applications*, 2019, 35(11): 102 – 105.
(徐虎. 基于改进萤火虫算法的铁路运行调度算法研究. 微型电脑应用, 2019, 35(11): 102 – 105.)
- [18] YU Shengping, CHAI Tianyou. Heuristic scheduling method for steelmaking and continuous casting production process. *Control Theory & Applications*, 2016, 33(11): 1413 – 1421.
(俞胜平, 柴天佑. 炼钢-连铸生产启发式调度方法. 控制理论与应用, 2016, 33(11): 1413 – 1421.)
- [19] GAO Yang, CHEN Shifu, LU Xin. Research on reinforcement learning technology: a review. *Acta Automatica Sinica*, 2004, 30(1): 86 – 100.
(高阳, 陈世福, 陆鑫. 强化学习研究综述. 自动化学报, 2004, 30(1): 86 – 100.)
- [20] LU Shoufeng, WEI Qiping, LIU Ximin. The study on off-line Q-learning model for single intersection signal timing. *Control Engineering of China*, 2012, 19(6): 987 – 992.
(卢守峰, 韦钦平, 刘喜敏. 单交叉信号配时的离线Q学习模型研究. 控制工程, 2012, 19(6): 987 – 992.)
- [21] HAO Wuwei, LI Junji. Study on the high dimensional problem optimization by particle swarm algorithm. *Control Engineering of China*, 2018, 25(5): 870 – 877.
(郝武伟, 李俊吉. 粒子群算法对高维问题的优化研究. 控制工程, 2018, 25(5): 870 – 877.)
- [22] SEMROV D, MARSETIC R, ZURA M, et al. Reinforcement learning approach for train rescheduling on a single-track railway. *Transportation Research-Part B*, 2016, 86: 250 – 267.

作者简介:

韩忻辰 硕士研究生, 主要从事深度强化学习与调度理论及应用等研究, E-mail: xinchenhan@126.com;

俞胜平 副教授, 硕士生导师, 主要从事调度理论及应用、全流程质量智能分析与预测等研究, E-mail: spyu@mail.neu.edu.cn;

袁志明 研究员, 主要从事行车指挥自动化、列车运行控制、智能调度与多列车协同控制等研究, E-mail: 13810696163@139.com;

程丽娟 硕士研究生, 主要从事列车运行调整算法研究, E-mail: 1377243675@qq.com.