控制与决策
Control and Decision

基于策略梯度强化学习的高铁列车动态调度方法

俞胜平, 韩忻辰, 袁志明, 崔东亮

引用本文:
俞胜平,韩忻辰,袁志明,崔东亮. 基于策略梯度强化学习的高铁列车动态调度方法[J]. 控制与决策, 2022, 37(9): 2407−2417.

在线阅读 View online: https://doi.org/10.13195/j.kzyjc.2021.0670

基于策略梯度强化学习的高铁列车动态调度方法

俞胜平[1†], 韩忻辰[1], 袁志明[2], 崔东亮[1]

(1. 东北大学 流程工业综合自动化国家重点实验室，沈阳 110004；
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: 高速铁路以其运输能力大、速度快、全天候等优势,取得了飞速蓬勃的发展.而恶劣天气等突发事件会导致列车延误晚点,更甚者延误会沿着路网不断传播扩散,其带来的多米诺效应将造成大面积列车无法按计划运行图运行. 目前依靠人工经验的动态调度方式难以满足快速优化调整的实际要求. 因此,针对突发事件造成高铁列车延误晚点的动态调度问题,设定所有列车在各站到发时间晚点总和最小为优化目标,构建高铁列车可运行情况下的混合整数非线性规划模型,提出基于策略梯度强化学习的高铁列车动态调度方法,包括交互环境建立、智能体状态及动作集合定义、策略网络结构及动作选择方法和回报函数建立,并结合具体问题对策略梯度强化学习(REINFORCE)算法进行误差放大和阈值设定两种改进. 最后对算法收敛性及算法改进后的性能提升进行仿真研究,并与Q-learning算法进行比较,结果表明所提出的方法可以有效地对高铁列车进行动态调度,将突发事件带来的延误影响降至最小,从而提高列车的运行效率.

关键词: 高铁列车；突发扰动；动态调度；强化学习；策略梯度；策略梯度强化学习

中图分类号: TP273 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2021.0670

引用格式: 俞胜平,韩忻辰,袁志明,等. 基于策略梯度强化学习的高铁列车动态调度方法[J]. 控制与决策, 2022, 37(9): 2407-2417.

A policy gradient reinforcement learning algorithm for high-speed railway dynamic scheduling

YU Sheng-ping[1†], *HAN Xin-chen*[1], *YUAN Zhi-ming*[2], *CUI Dong-liang*[1]

(1. State Key Laboratory of Synthetical Automation for Process Industries，Northeastern University，Shenyang 110004，China；2. Signal & Communication Reseach Institute，China Academy of Railway Sciences Co., Ltd，Beijing 100081，China)

Abstract: The high-speed railway has achieved vigorous development in recent years due to its advantages of large transport capacity, fast speed and all-weather. But unexpected events such as bad weather will cause train delays, and even the delay will continue to spread along the road network. The domino effect will cause large-area trains to fail to operate according to the plan. At present, the dynamic scheduling method relying on manual experience is difficult to meet the actual requirements. Therefore, this paper aims at the problem of dynamic scheduling of high-speed train, setting the minimum sum of the delays of all trains at each station as the optimization goal. At the same time, a mixed-integer nonlinear programming (MINLP) model under traversable conditions is constructed, and a policy gradient reinforcement learning method is proposed including establishment of environment, definition of state and action set, policy network, action selection method, reward function and combined with the specific problems, the error amplification and threshold setting of REINFORCE algorithm are improved. Finally, the convergence and the performance improvement of the algorithm are studied and compared with the Q-learning algorithm. The results show that the method proposed in this paper can effectively reschedule high-speed trains, minimize the impact of delays, and improve the efficiency of train operation.

Keywords: high-speed railway；unexpected disturbances；dynamic scheduling；reinforcement learning；policy gradient；REINFORCE

收稿日期: 2021-04-20；**录用日期:** 2021-06-17.

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

责任编委: 刘民.

[†]通讯作者. E-mail: spyu@mail.neu.edu.cn.

0 引 言

作为我国综合交通运输体系的骨干核心,高速铁路以其运输能力大、速度快、全天候等优势,近些年来取得了飞速蓬勃的发展. 随着高铁数量的不断增多与路网结构的愈加复杂化,两者之间存在的强耦合、快演变、多约束等特点越来越明显. 恶劣天气、列车故障等突发事件可能会导致单列车或多列车出现延误晚点的现象,更甚者延误会沿着路网不断传播扩散,其带来的多米诺效应将造成大面积列车无法按计划运行图运行. 因此,当列车运行受到突发事件影响以致偏离初始调度计划时,如何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,如何对受影响列车进行高效准确地动态调度,从而实现高铁列车运行快速、运营安全、到站准时就显得尤为重要.

高铁列车动态调度问题是一类包含多目标、非线性约束的优化问题,针对此问题已有的研究方法大致可以分为仿真方法、运筹学方法、启发式方法以及机器学习方法.

通过仿真方法能够对系统有一个较为直观的展示[1-2]. 针对列车图像控制操作的驾驶规则,文献[3]建立了一个高铁列车仿真模型,包括3D建模、视频处理等;文献[4]在经典的列车数字仿真模型上进行了改进,并设计了简单的用户图形界面;文献[5]建立了一个纵向经线的列车动力学模型,为列车微观仿真提供了良好的理论基础.

最优化方法作为一种精确求解方法,可保证求解问题的全局优化[6-8]. 文献[9]根据原计划时刻表所受影响的不同程度,提出了Ideal、Quasi-ideal以及Feasible Solutions三种解决方案;文献[10]采用自适应动态规划体系中的双重启发式动态规划算法,对列车晚点的调整起到了良好的控制作用.

启发式方法利用面向特定问题的知识和经验,可以产生比较合适的解决方法[11-13]. 文献[14]提出了3种切换策略No-SP、Original-SP以及Improved-SP,并在真实数据仿真结果下,说明了改进算法的优越性;文献[15-16]均采用改进的粒子群优化算法对受扰动列车群进行动态调度.

运用机器学习方法,尤其是强化学习方法解决高铁动态调度问题的研究目前相对较少. 文献[17]采用强化学习中经典的Q-learning算法进行列车调度,是第一篇运用强化学习方法进行铁路列车调度研究的文章;文献[18]通过改进Q-learning算法中的状态空间、动作策略及奖励系统等,将改进后的算法应用于路网可扩展为大规模且双向的情形;文献[19]在文献[18]的基础上对状态向量、动作奖励及仿真环境等进行了改进,并对解的质量进行了对比. 同样,针对受扰动列车调度问题,文献[20-21]采用一种深度强化学习(deep Q-network)的方法,用以解决列车调度问题.

采用仿真方法的结果较为直观,但当情况复杂时,仿真的效果会随之下降. 最优化方法缺乏实时性和适应性,不能很好地满足复杂情况下的列车动态调度需求. 启发式方法,尤其以智能优化算法为主,在求解多约束、多变量问题时,迭代计算次数多,且容易陷入局部最优. 强化学习方法对高铁列车动态调度问题决策优化的优势便体现出来. 首先,强化学习是以目标为导向的学习工具,智能体通过与环境交互学习来最优化需要的策略,若目标明确,环境设置合理,则智能体可很快学习到最优调度策略. 其次,强化学习方法中的智能体会调整探索-利用之间的平衡,以最大程度地避免在训练开始阶段就过多采取某一非最优策略.

本文将选取双线双向铁路的其中一条单线单向多车站列车线路为仿真场景,对突发事件造成高铁列车延误晚点的动态调度问题开展研究. 首先,建立以最小化列车在各站到发时间晚点总和为目标函数的混合整数非线性规划模型,并据此模型建立强化学习中智能体用于交互学习的环境. 在此基础上,对策略梯度中的REINFORCE算法采用误差放大与设定阈值两种改进技巧. 最终仿真结果表明,本文提出的REINFORCE算法可以有效地解决高铁列车的动态调度问题,得到满意的动态调度策略,从而可以为高铁列车的优化调度运行提供良好的决策依据.

1 问题描述

高铁列车运行安排是按照高铁部门制定的计划运行图来执行的. 计划运行图包含了每辆列车的行进线路、在线路上的经停车站,以及在各个站点的到站时刻和离站时刻. 正常情况下,高铁列车按照计划运行图有序高效运行.

因大风、暴雪或列车故障等突发事件导致列车晚点是铁路运输组织中的一种常见现象. 据统计,我国高速铁路2015年正点率不足90%[22]. 因此,研究晚点情况下的高铁列车动态调度问题具有重要意义. 本文针对突发事件导致受影响列车无法按照计划运行图运行的动态调度问题展开研究,研究对象选取复线铁路中的单线单向多车站场景.

目前,对于列车晚点现象,更多的是依靠调度员

经验按列车逐站调整或者整体平移列车运行时分的人工调整方式. 这种调整方式前瞻性较差,在错综复杂的路网条件下将更难进行,从而不能将延误影响最小化. 因此,在本文所设定的场景和约束条件的基础上,建立高铁列车动态调度数学模型,采用改进后的REINFORCE算法,对发生晚点的列车及其连带影响晚点的列车进行动态调度,将突发事件带来的延误影响降至最小.

2　高铁列车动态调度模型

2.1　参数定义

参数及其含义设置如下:

Ω^0:计划运行图中的所有列车集合.

Ω:需动态调度的列车集合,$|\Omega|$表示集合中列车总数量.

N:需动态调度的列车数量,$N=|\Omega|$.

i,i_1,i_2:列车索引号,$i,i_1,i_2=1,2,\dots,N$,按照扰动发生时Ω中列车在线路运行中的先后顺序进行索引标号.

j:车站索引号,$j=1,2,\dots,M$,M表示线路中车站总数量. 按车站从始发站到终点站的前后顺序进行索引标号.

i^*:突发事件直接导致发生离站晚点的列车索引.

j^*:突发事件直接导致列车i^*发生离站晚点的车站索引.

$T^*_{i^*j^*}$:列车i^*在车站j^*的离站延误时间.

R_j:车站j的股道集合,$|R_j|$表示车站j的股道总数量.

R:所有股道集合,$R=\{R_1,\dots,R_j,\dots,R_M\}$.

l:股道索引号.

q:两站之间区段的索引号,$q=1,2,\dots,Q$,Q表示区段的总数量.

B_q:第q个区段的闭塞区间集合,$|B_q|$表示第q个区段内闭塞区间总数量.

B_q^{first}:第q个区段里的第1个闭塞区间.

B_q^{last}:第q个区段里的最后1个闭塞区间.

B:闭塞区间集合,$B=\{B_1,\dots,B_q,\dots,B_Q\}$.

k:闭塞区间索引号.

A^0_{ij}:计划运行图中,列车i到达车站j的计划到站时间.

D^0_{ij}:计划运行图中,列车i离开车站j的计划离站时间.

A^{Δ}_{ij}:列车i在车站j的实际到站时间.

D^{Δ}_{ij}:列车i在车站j的实际离站时间.

T^a_j:在车站j,相邻列车之间最小到站间隔时间.

T^d_j:在车站j,相邻列车之间最小离站间隔时间.

$T^{\min}_{ij}$:列车i在车站j的最小停站时间.

$T^{\min}_{i,j,j+1}$:列车i在$j,j+1$站之间最小区间运行时间.

η_{ij}:列车i在车站j的状态参数. 若列车i在车站j未到站,则$\eta_{ij}=0$;若列车i在车站j正停靠,则$\eta_{ij}=1$;若列车i在车站j已离站,则$\eta_{ij}=2$.

b_{kt}:闭塞区间k在时刻t包含的列车数量.

r_{lt}:股道l在时刻t包含的列车数量.

U:一个非常大的正实数.

T:离散时间点的总数量.

决策变量及其含义设置如下:

A_{ij}:列车i在车站j的到站时间.

D_{ij}:列车i在车站j的离站时间.

$x_{i_1,i_2,j}$:列车i_1和列车i_2在车站j的到站顺序变量. 若i_1先于i_2在车站j到站,则$x_{i_1,i_2,j}=1$;否则$x_{i_1,i_2,j}=0$.

$y_{i_1,i_2,j}$:列车i_1和列车i_2在车站j的离站顺序变量. 若i_1先于i_2在车站j离站,则$y_{i_1,i_2,j}=1$;否则$y_{i_1,i_2,j}=0$.

$z_{i_1,i_2,j}$:列车i_1和列车i_2在车站j的越行变量. 若i_1和i_2在车站j发生越行,则$z_{i_1,i_2,j}=1$;否则$z_{i_1,i_2,j}=0$.

2.2　优化目标

当突发事件发生并造成列车晚点时,为满足旅客出行对准时性的要求,应尽量减小各列车在各站的晚点时间,尽快恢复各列车的准点运行. 故构建目标函数如下:

$$\text{Min}\ \ F=\sum_{i=1}^{N}\sum_{j=1}^{M-1}(D_{ij}-D^0_{ij})+\sum_{i=1}^{N}\sum_{j=2}^{M}|A_{ij}-A^0_{ij}|. \tag{1}$$

2.3　约束条件

1) 列车在中间经停车站的停站时间不得小于车站规定的最小停站时间约束,即

$$D_{ij}-A_{ij}\geqslant T^{\min}_{ij},\quad i=1,2,\dots,N,\ j=2,3,\dots,M-1. \tag{2}$$

2) 列车离站时间不得早于列车计划离站时间约束,即

$$D_{ij}\geqslant D^0_{ij},\ i=1,2,\dots,N,\ j=2,3,\dots,M-1. \tag{3}$$

3) 相邻列车在车站的最小进站时间间隔约束,即

$$(A_{i_2,j}-A_{i_1,j})+U(1-x_{i_1,i_2,j})\geqslant T_j^a,$$
$$i_1,i_2=1,2,\ldots,N,\ i_1\neq i_2,\ j=2,3,\ldots,M. \quad (4)$$

4) 相邻列车在车站的最小离站时间间隔约束,即

$$(D_{i_2,j}-D_{i_1,j})+U(1-y_{i_1,i_2,j})\geqslant T_j^d,$$
$$i_1,i_2=1,2,\ldots,N,\ i_1\neq i_2,\ j=1,2,\ldots,M-1. \quad (5)$$

5) 列车在车站之间的最小区间运行时分约束,即

$$A_{i,j+1}-D_{ij}\geqslant T_{i,j,j+1}^{\min},$$
$$i=1,2,\ldots,N,\ j=1,2,\ldots,M-1. \quad (6)$$

6) 同一个闭塞区间在任何时刻最多只允许一辆列车单独运行,即

$$b_{kt}\leqslant 1,\ k\in B,\ t=1,2,\ldots,T. \quad (7)$$

7) 车站股道最多只允许一辆列车占用,即

$$r_{lt}\leqslant 1,\ l\in R,\ t=1,2,\ldots,T. \quad (8)$$

8) 列车在车站之间线路区段不发生越行,即

$$x_{i_1,i_2,j}=y_{i_1,i_2,j-1},$$
$$i_1,i_2=1,2,\ldots,N,\ i_1\neq i_2,\ j=2,3,\ldots,M. \quad (9)$$

9) 列车在已到车站的到站时间约束,即

$$A_{ij}=A_{ij}^{\Delta},$$
$$i=1,2,\ldots,N,\ j=2,3,\ldots,M-1,\ \eta_{ij}=1. \quad (10)$$

10) 列车在已离车站的离站时间约束,即

$$D_{ij}=D_{ij}^{\Delta},$$
$$i=1,2,\ldots,N,\ j=1,2,\ldots,M-1,\ \eta_{ij}=2. \quad (11)$$

11) 变量取值约束,即

$$A_{ij}\geqslant 0,\ i=1,2,\ldots,N,\ j=2,3,\ldots,M. \quad (12)$$
$$D_{ij}\geqslant 0,\ i=1,2,\ldots,N,\ j=1,2,\ldots,M-1. \quad (13)$$
$$x_{i_1,i_2,j}=\begin{cases}1,\ A_{i_1,j}<A_{i_2,j};\\ 0,\ \text{otherwise}.\end{cases}$$
$$i_1,i_2=1,2,\ldots,N,\ i_1\neq i_2,\ j=2,3,\ldots,M. \quad (14)$$
$$y_{i_1,i_2,j}=\begin{cases}1,\ D_{i_1,j}<D_{i_2,j};\\ 0,\ \text{otherwise}.\end{cases}$$
$$i_1,i_2=1,2,\ldots,N,\ i_1\neq i_2,\ j=1,2,\ldots,M-1. \quad (15)$$
$$z_{i_1,i_2,j}=\begin{cases}0,\ x_{i_1,i_2,j}=y_{i_1,i_2,j};\\ 1,\ x_{i_1,i_2,j}\neq y_{i_1,i_2,j}.\end{cases}$$
$$i_1,i_2=1,2,\ldots,N,\ i_1\neq i_2,\ j=1,2,\ldots,M-1. \quad (16)$$

2.4 高铁列车动态调度数学模型

根据上述优化目标及各约束条件,建立如下高铁列车动态调度数学模型:

$$\text{object}: \text{式}(1),$$
$$\text{s.t. 式}(2)\sim(16).$$

上述建立的模型为混合整数非线性规划模型,包含大量约束方程、离散变量和连续变量. 而强化学习与深度学习结合后可解决高维数据的输入和计算问题,以试错寻优的机制不断进行改进,从而得到满意的动态调度策略. 因此,本文将采取强化学习中REINFORCE方法对上述模型进行求解.

3 基于REINFORCE的高铁动态调度

策略梯度方法需要显式地表示策略函数,通过梯度优化的方法持续改善目标函数,从而使得策略也随之改进,其中梯度方向是使得策略改进最快的方向. 相比于值函数方法,策略梯度方法既可以处理确定性策略,又可以处理随机性策略,并在理论上能够保证算法收敛.

3.1 交互环境

在建立交互环境之前,首先提出4个为方便处理问题的假设条件.

假设1 假设在交互环境中运行的列车为一个质点,即忽略列车的长度.

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.

假设3 假设在车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代替.

假设4 假设每辆列车只有两个动作前进或者停止,并忽略列车加减速过程.

根据上述提出的4个假设,结合2.4节的动态调度数学模型,建立实验的交互环境. 首先对环境中可占用资源H进行说明,其包括车站内的股道资源R及车站间的区段资源B. 对于车站内的股道资源,按照列车运行始发站至终点站的顺序编号$R=[R_1,R_2,\ldots,R_j,\ldots R_M]$,其表示共有$M$个车站. 对于车站间的区段资源,按照线路中的前后位置进行编号$B=[B_1,B_2,\ldots,B_q,\ldots,B_Q]$,其表示共有$Q$个区段,这里$|B_q|$表示第$q$个区段内的闭塞区间总个数. 在环境资源中,车站资源与区段资源交替出现,即

$$H=[R_1,B_1,\ldots,R_q,B_q,\ldots,R_Q,B_Q,R_{Q+1}].$$

为了描述列车占用的资源情况,本文建立列车状态矩阵S,用来保存各列车在各个时刻所占用的资源情况,以满足2.4节数学模型中的所有约束条件. 列车状态矩阵S具体形式如下:

$$S=[s_{i,t}]_{N\times T}=\begin{bmatrix} s_{1,1} & \cdots & s_{1,t} & \cdots & s_{1,T} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ s_{i,1} & \cdots & s_{i,t} & \cdots & s_{i,T} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ s_{N,1} & \cdots & s_{N,t} & \cdots & s_{N,T} \end{bmatrix}. \tag{17}$$

其中: S中的第i行表示列车i,第t列表示时刻t; S中的每个元素$s_{i,t}$表示在第t时刻列车i占用的资源编号; N表示需要进行动态调度的列车总数; T表示动态调度的时间跨度; $s_{\cdot,t}$表示将S按列分块后的第t列元素.

3.2　智能体状态及动作集合的定义

针对路网及各列车之间存在的强耦合性,本文将突发事件影响到的所有列车看作一个列车群,令其作为智能体进行调度策略的优化. 根据本文建立的交互环境,将智能体状态定义为在t时刻各列车占用的资源编号,即智能体的状态包含时间和空间两个维度的信息,如下所示:

$$\text{state}_t=[h_1,\ldots,h_i,\ldots,h_N;t]. \tag{18}$$

其中: t表示第t时刻; h_i表示列车i占用的资源编号. 根据上述状态定义可发现state_t与S中元素的关系为$\text{state}_t=[[s_{\cdot,t}]^{\mathrm{T}};t]_{1\times(N+1)}$.

对于列车i,在t时刻可选择的动作共两种,前进(1)或停止(0),即$\text{act}_{i,t}\in\{0,1\}$. 根据上述智能体的定义,在$t$时刻,智能体可采取的动作集合如下,其对应执行的动作有2^N种可能:

$$\text{action}_t=\{\text{act}_{1,t},\ldots,\text{act}_{i,t},\ldots,\text{act}_{N,t}\}. \tag{19}$$

根据上述智能体状态及动作定义可发现其均是以交互环境中最小决策时间为单位进行更新. 所以当扰动发生在t'时,智能体可在同一时刻做出相适的动态调度策略,以最小化扰动带来的影响.

3.3　策略网络结构及动作选择方法

1989年, Hornik等[23]提出的“万能近似定理”为深度学习提供了理论依据. 这种依靠大量神经元细胞连接成的神经网络,可建立输入数据与输出数据的复杂映射关系,其自学习的能力常用来解决非线性复杂系统的建模问题. 结合本文建立的模型及交互环境,确定策略网络基本结构如图1所示.

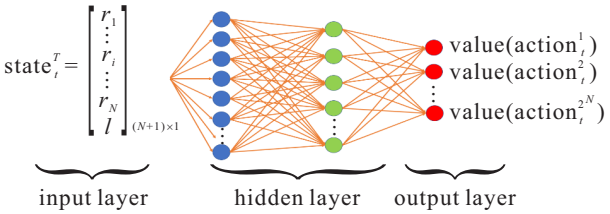


图 1　策略网络

图1中,策略神经网络的输入层为智能体在t时刻的状态state_t^T,经过隐含层的计算,在输出层得到该state_t^T下智能体所有动作的值value. 可以确定输入层的维数为$N+1$维,输出层维度为2^N维.

本文动作选取方法是在策略网络输出各动作的值value之后,经过一个softmax操作,如图2所示.

将value(a_i)转化为每个动作对应的概率值pro(a_i),根据pro(a_i)对动作进行采样执行. 经过softmax网络后可以更直观地体现出每个动作被选中的概率大小. 但是,需要注意在选择动作时并非是对2^N个动作都进行选择,而是根据之前交互环境中约束条件的判断,只在满足约束的动作集合中按pro(a_i)进行选择,以此保证智能体最终选择的动作序列是满足前述约束条件的,即产生的是满足约束的可行解.

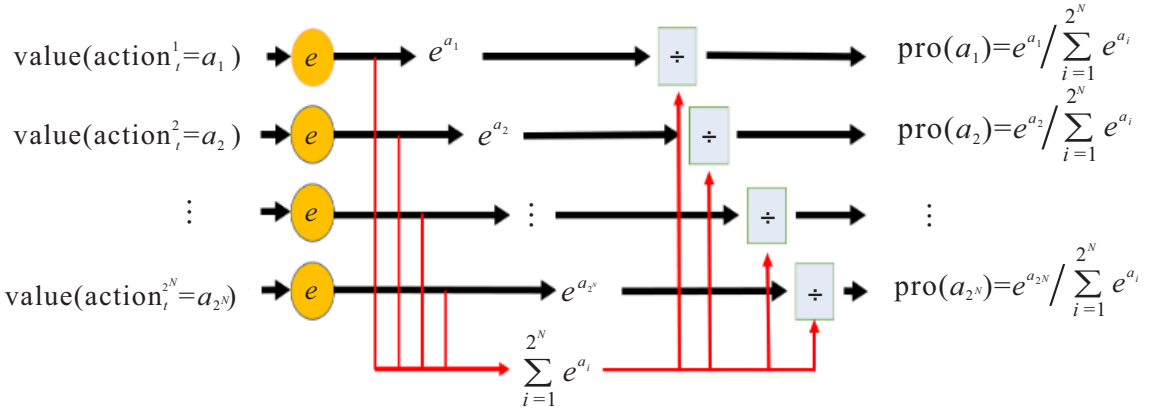


图 2　softmax网络

3.4 回报函数

根据本文建立数学模型的优化目标,智能体在完成一轮完整回合后的策略评估函数按下式进行计算,对此轮的策略网络参数进行评价及改进:

$$G=\frac{1}{F}. \tag{20}$$

由式(20)可以看出,列车在各站的到发延迟时间总和越小,给予智能体的回报越大,对于当前策略的评价越高. 利用式(20)作为回报函数容易出现以下两个问题:

问题1及改进方法:对于相差较小的目标函数值,策略网络参数对当前输出策略的评价不敏感,容易陷入局部最优. 对于此问题,本文采取误差放大的改进方法,即对式(20)放大10 000倍. 这样处理使得网络在训练时,可以更好地区分目标函数值较为接近的情况,使得网络参数可以更好地训练.

问题2及改进方法:初始化策略网络参数后,其对于初始解的求解用时过长,且计算得到初始解的效果也并不显著,导致后续在该解的基础上利用REINFORCE算法进行优化费时较长. 对于此问题,本文采取设定阈值的方法,设定一个动态调度后的时间跨度阈值$T_{\rm threshold}$. 对于目标函数在$T_{\rm threshold}$之上的解,认为其性能较劣,不利用其进行网络参数的更新,而是让智能体继续与环境交互学习. 只有采样到较优的目标函数解时,才在此解的基础上对网络参数进行更新.

综上所述,最终回报函数G更新如下:

$$G=\begin{cases}\dfrac{10\,000}{F},\ \text{object}\leqslant T_{\rm threshold};\\ \dfrac{1}{M},\ \text{otherwise}.\end{cases} \tag{21}$$

其中M表示一非常大的正整数.

3.5 算法流程

step 1: 定义学习率α、折扣因子β,导入计划运行图对应的列车时刻表T-table,根据T-table信息构建策略网络模型model,确定扰动作用点delay_point、延误时间delay_time,初始化环境、智能体及其状态、列车状态矩阵S、总训练回合数max_episodes,构建记录本回合内所经历过的状态、动作列表state_buffer和action_buffer.

step 2(for loop): 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$t=0$,初始化智能体的起始状态state.

step 3(while loop): 将当前时刻的t与state进行压缩,变为input_state,输入model得到输出output_action,对其做softmax处理并仅保留满足约束的动作,后进行采样得到真正需要执行的动作action.

step 4: 根据t、state、action得到下一时刻的状态state_以及本轮训练是否结束的标志done. 同时将state、action存入到state_buffer和action_buffer中.

step 5: 判断本回合是否结束. 如果否,则回到step 3,继续本轮episode内的采样;如果是,则此时已经获得回报return,本轮训练结束.

step 6: 本轮训练结束后,首先对action_buffer中的每个元素进行one-hot编码并计算交叉熵,即

$$\begin{aligned}&\text{cross\_entropy}[i]=\\&-\sum_j \text{model}[i][j]\times\log(\text{prob\_action}[i][j]).\end{aligned} \tag{22}$$

然后将本轮得到的return采用折扣型奖励的方式计算得到每一步奖励reward,将每一步reward存入reward_buffer,并对其进行标准化处理. 最终得到策略网络的损失函数loss,即

$$\text{loss}=\frac{1}{n}\sum_{i=1}^{n}\text{cross\_entropy}[i]\times\text{reward\_buffer}[i]. \tag{23}$$

step 7: 对loss进行梯度下降,更新网络参数.

step 8: episode+1,如果智能体最后结果已经收敛或者已经达到max_episodes,则退出训练并输出经过训练优化后的调度计划opt_plan,否则回到step 2.

4 仿真验证

4.1 交互环境与算法有效性验证

为验证本文交互环境的合理性,设计具体的三车三站两区间、四车五站四区间、十车十站九区间3种不同的仿真场景,各场景下列车计划运行时刻表如表1～表3所示.

表1 场景1计划时刻表

车次	车站1	车站2		车站3
	发	到	发	到
G1	11:03	11:13	11:16	11:19
G2	11:06	11:16	11:19	11:22
G3	11:10	11:20	11:23	11:26

表2 场景2计划时刻表

车次	车站1	车站2		车站3		车站4		车站5
	发	到	发	到	发	到	发	到
G1	8:32	8:42	8:45	8:54	8:57	9:01	9:04	9:08
G2	8:40	8:50	8:53	9:02	9:05	9:09	9:12	9:16
G3	8:45	8:55	8:58	9:07	9:10	9:14	9:17	9:21
G4	8:51	9:01	9:04	9:13	9:16	9:20	9:23	9:27

表 3　场景3计划时刻表

车次	车站1	车站2		车站3		车站4		车站5		车站6		车站7		车站8		车站9		车站10
	发	到	发	到	发	到	发	到	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	11:50	11:58	12:01	12:03	12:06	12:14	12:17	12:19
G2	11:06	11:16	11:19	11:25	11:28	11:30	11:33	11:41	11:44	11:50	11:53	12:01	12:04	12:06	12:09	12:17	12:20	12:22
G3	11:10	11:20	11:23	11:29	11:32	11:34	11:37	11:45	11:48	11:54	11:57	12:05	12:08	12:10	12:13	12:21	12:24	12:26
G4	11:15	11:25	11:28	11:34	11:37	11:39	11:42	11:50	11:53	11:59	12:02	12:10	12:13	12:15	12:18	12:26	12:29	12:31
G5	11:20	11:30	11:33	11:39	11:42	11:44	11:47	11:55	11:58	12:04	12:07	12:15	12:18	12:20	12:23	12:31	12:34	12:36
G6	11:23	11:33	11:36	11:42	11:45	11:47	11:50	11:58	12:01	12:07	12:10	12:18	12:21	12:23	12:26	12:34	12:37	12:39
G7	11:28	11:38	11:41	11:47	11:50	11:52	11:55	12:03	12:06	12:12	12:15	12:23	12:26	12:28	12:31	12:39	12:42	12:44
G8	11:32	11:42	11:45	11:51	11:54	11:56	11:59	12:07	12:10	12:16	12:19	12:27	12:30	12:32	12:35	12:43	12:46	12:48
G9	11:37	11:47	11:50	11:56	11:59	12:01	12:04	12:12	12:15	12:21	12:24	12:32	12:35	12:37	12:40	12:48	12:51	12:53
G10	11:42	11:52	11:55	12:01	12:04	12:06	12:09	12:17	12:20	12:26	12:29	12:37	12:40	12:42	12:45	12:53	12:56	12:58

在每种场景下各仿真3种不同的延误晚点情况,据此来说明动态调度策略的有效性. 具体各场景仿真参数如表4～表6所示. 表4～表6中: 车站列对应的数字表示该车站内的股道数,括号内数字表示列车在该站的最小停站时间;区间列对应的数字表示该区间内的闭塞区间数,括号内数字表示列车在某一闭塞区间内的最小运行时间.

为更直观地理解表4～表6内参数对应的实际运行线路,将场景2的仿真参数可视化后如图3所示.

表 4　场景1具体参数

车站1	区间1	车站2	区间2	车站3	总资源数
3	5(2)	2(3)	3(1)	3	16

表 5　场景2具体参数

车站1	区间1	车站2	区间2	车站3	区间3	车站4	区间4	车站5	总资源数
4	5(2)	3(3)	3(3)	3(3)	4(1)	3(3)	2(2)	4	31

表 6　场景3具体参数

车站1	区间1	车站2	区间2	车站3	区间3	车站4	区间4	车站5	区间5
10	5(2)	6(3)	3(2)	7(3)	2(1)	6(3)	4(2)	8(3)	3(2)
车站6	区间6	车站7	区间7	车站8	区间8	车站9	区间9	车站10	总资源数
10(3)	4(2)	8(3)	2(1)	6(3)	4(2)	8(3)	2(1)	10	108

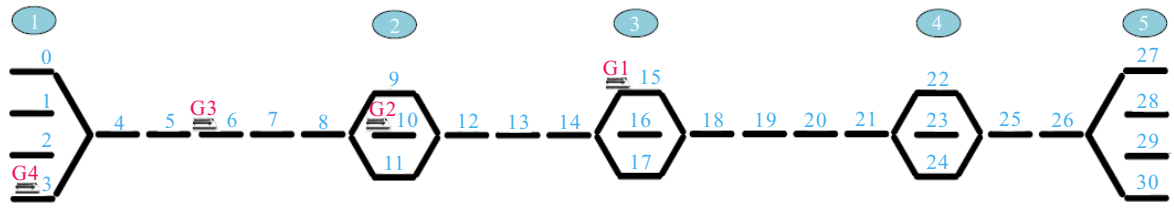


图 3　资源分布

规定列车速度为300 km/h. 对于2.4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如表4～表6中数据所示,并规定经停站最小作业时间$T_{ij}^{\min} = 3$ min. 为确定每个场景在每种扰动下的阈值,需要计算每个场景第1辆列车从首发车站发车至最后一辆列车到最终车站的时间跨度T_{span},其中场景1～场景3的T_{span}分别如下所示:

$$T_{\text{span\_1}} = 11{:}26 - 11{:}03 = 23\,\text{min}, \tag{24}$$

$$T_{\text{span\_2}} = 9{:}27 - 8{:}32 = 55\ \text{min}, \tag{25}$$

$$T_{\text{span\_3}} = 12{:}58 - 11{:}03 = 115\ \text{min}. \tag{26}$$

阈值设定为

$$T_{\text{threshold}} = T_{\text{span}} + T_{i^*j^*}^* + \lceil 0.5 \times T_{\text{span}} \rceil, \tag{27}$$

其中$\lceil\cdot\rceil$表示上取整符号.

具体各场景下的扰动晚点列车、扰动晚点发生所在车站及晚点时间如表7所示. 表7的$T_{\text{threshold}}$列根据式(27)计算所得,表7的最后一列为各晚点情况下经本文所提算法求得的目标函数值. 由此发现,在多种场景、多种扰动下,算法对受扰动列车的动态调

度是相适且有效的.

表 7 各场景下不同扰动

扰动作用点	车站	列车	时间 (min)	$T_{\text{threshold}}$ (min)	目标函数 (min)
场景 1	车站 1	G1	5	40	20
	车站 1	G2	7	42	28
	车站 2	G1	10	45	20
场景 2	车站 1	G3	10	93	80
	车站 2	G1	16	99	96
	车站 2	G2	25	108	150
场景 3	车站 2	G1	15	188	240
	车站 3	G3	24	197	336
	车站 8	G2	40	213	160

经过实验确定超参数学习率$\alpha = 0.01$、折扣因子$\beta = 0.95$、$M = 10\,000\,000$,具体策略网络参数为

$$\text{hidden\_layer1}\begin{cases}\text{units} = 128,\\ \text{act} = \tanh;\end{cases}.$$

$$\text{hidden\_layer2}\begin{cases}\text{units} = 32,\\ \text{act} = \text{relu};\end{cases}.$$

$$\text{output\_layer}\begin{cases}\text{units} = 2^N,\\ \text{act} = \text{none}.\end{cases}$$

具体各场景训练结果如图4所示.

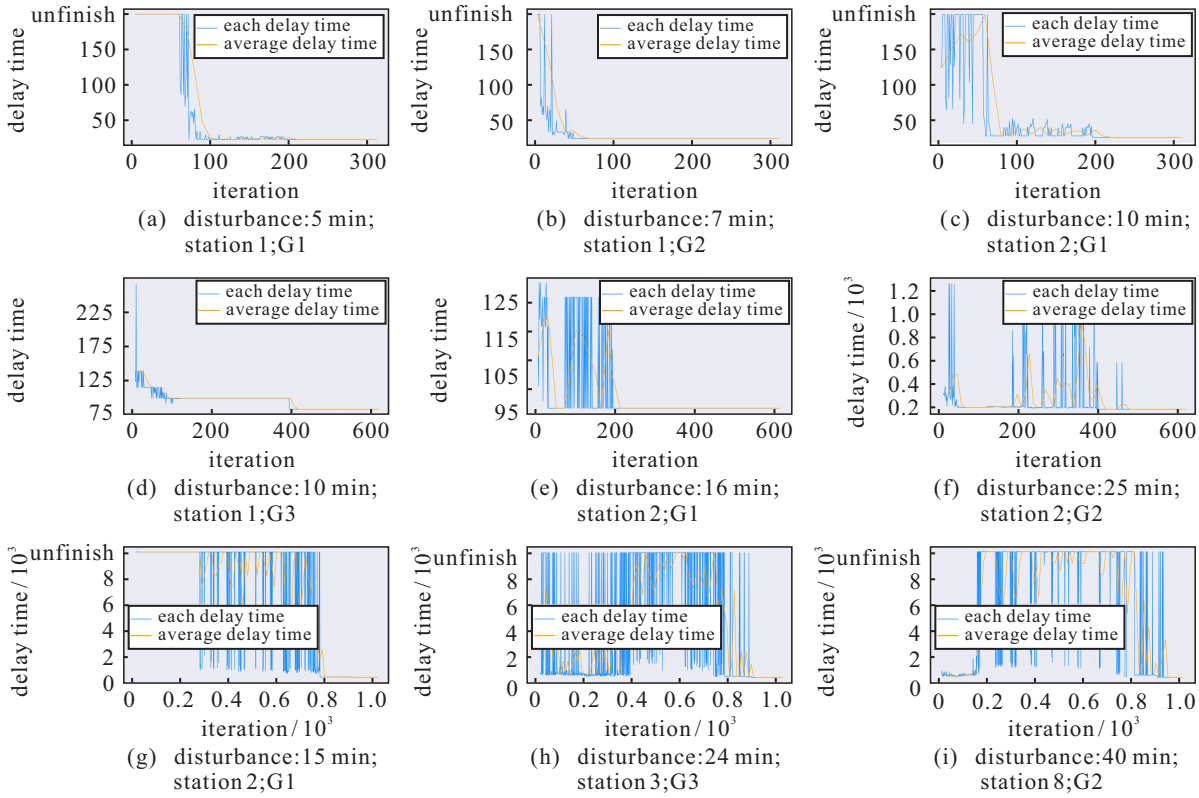


图 4 不同场景下的训练结果

图4训练结果显示,本文设计的调度模型、交互环境及回报函数是合理的,并且运用改进REINFORCE算法可以对该模型进行有效求解,在较少的训练回合内即收敛至最优解,可满足动态调度实时性需求.

4.2 改进REINFORCE算法回报函数的仿真比较

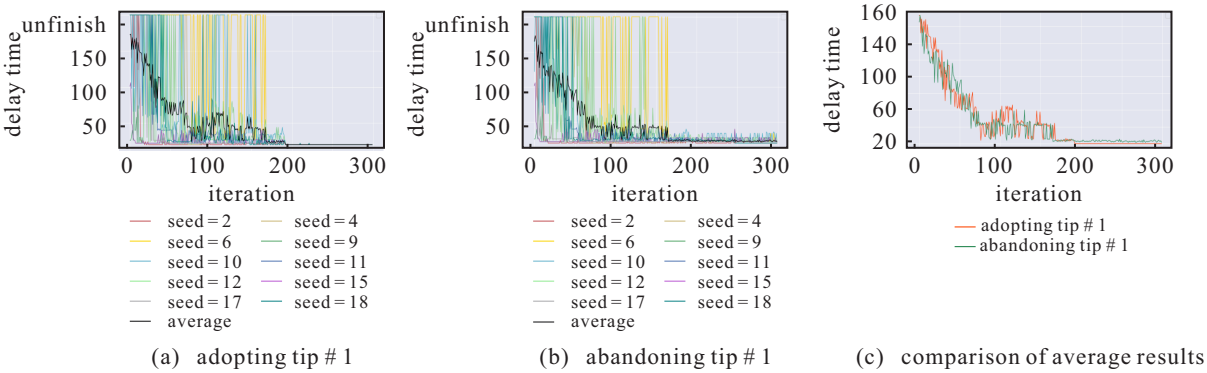
在3.4节中,针对智能体训练过程中出现的两个关键问题,分别给出了各自的解决方案,以下将通过仿真对比改进回报函数与未改进回报函数对REINFORCE算法性能的影响.

为比较改进后回报函数对算法性能优越性的提升,将从0～20中随机抽取10个随机数种子进行实验对比.

1) 针对问题1及改进方法,选定4.1节场景1的第3种扰动情况$T^*_{\text{G1,sta}_2} = 10\,\text{min}$, $T_{\text{threshold}} = 45\,\text{min}$. 若不采用误差放大方法,即回报函数在原基础上不做放大10 000倍处理,则可得回报函数如下:

$$G = \begin{cases}\dfrac{1}{F}, & \text{object} \leqslant T_{\text{threshold}};\\ \dfrac{1}{M}, & \text{otherwise}.\end{cases} \tag{28}$$

训练结果如图5所示. 图5(a)表示回报函数采用误差放大的改进方法,图5(b)表示没有采用该改进方法,图5(a)和图5(b)中黑色曲线均表示算法在10个随机数种子影响下所得结果取平均值的仿真曲线. 图5(c)将图5(a)和图5(b)中的平均值曲线单独进行比较,通过图5(c)可以发现在训练的开始及前半程两曲



(a) adopting tip # 1　(b) abandoning tip # 1　(c) comparison of average results

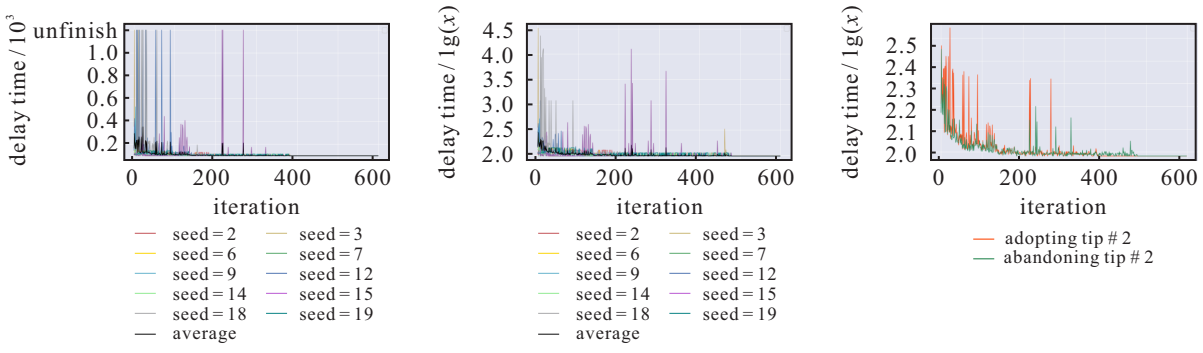
图 5　有无误差放大改进的结果对比

线的下降速率基本相同,但是在训练的后半程,未采用误差放大改进的绿色曲线仍在震荡,出现对近似解不敏感的现象,而采用改进后的红色曲线则避免了该问题.

2) 针对问题2及改进方法,选定4.1节场景2的第2种扰动情况 $T_{\mathrm{G1,sta_2}} = 16\,\mathrm{min}, T_{\mathrm{threshold}} = 99\,\mathrm{min}$,不采用阈值设定的改进方法,即不设定解的阈值下限. 所以回报函数如下:

$$G = \frac{10\,000}{F}. \tag{29}$$

训练结果如图6所示. 图6(a)表示回报函数采用阈值设定的改进方法,图6(b)表示未采用该改进方法. 因不采用阈值设定的改进方法智能体可能会长时间不采取前进的动作,导致延误时间数值特别大,为更好地展示训练效果,特将图6(b)和图6(c)的纵轴采用以10为底的对数坐标. 图6(a)和图6(b)中黑色曲线均表示在10个随机数种子影响下所得结果取平均值的仿真曲线. 图6(c)将图6(a)和图6(b)中的平均值曲线单独进行比较,通过图6(c)可以发现,采用阈值方法改进的智能体在训练前半程并不如未采用改进的智能体,原因是当人为地把阈值之上的解舍弃掉时,会使得智能体的学习不全面,但是这样可以保证其在较优解的基础上进行学习,所以在训练的后半程采用改进方法的智能体更早地收敛到最优解. 并且两种回报函数下的计算时间也存在较大差异,如表8所示.



(a) adopting tip # 2　(b) abandoning tip # 2　(c) comparison of average results

图 6　有无设定阈值改进的结果对比

表8　有无阈值设定改进的计算时间对比　　单位: s

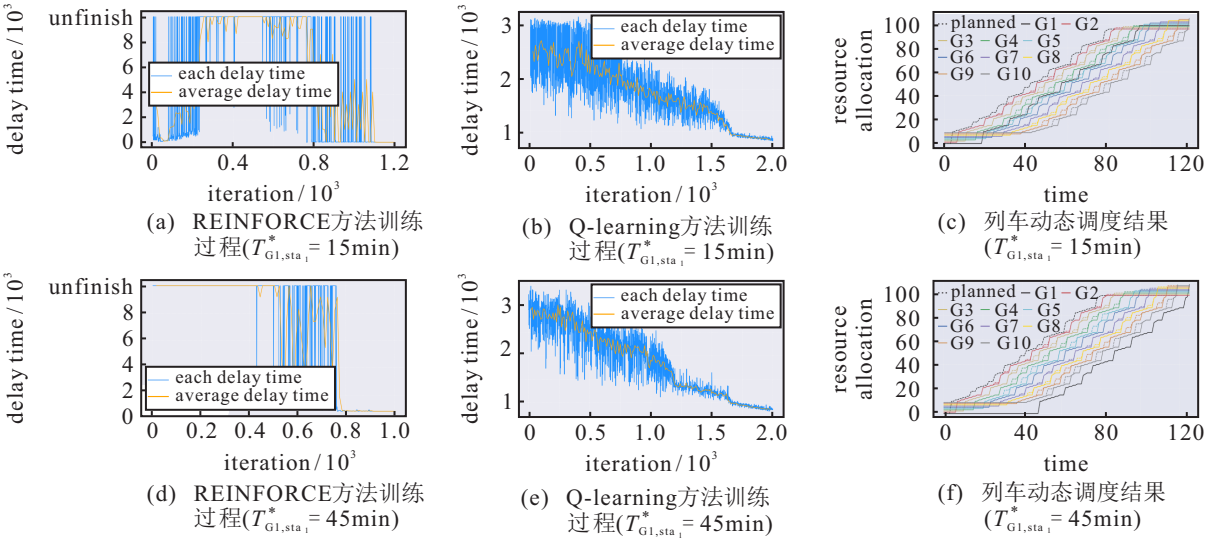
	seed										average
	2	3	6	7	9	12	14	15	18	19	
adopting tip 2	45.44	46.35	45.42	45.71	45.71	46.31	45.66	46.18	45.81	45.57	45.816
abandoning tip 2	44.93	52.10	55.25	55.24	46.04	52.42	45.69	85.53	128.5	45.92	61.162

4.3　REINFORCE与Q-learning算法比较

Q-learning算法也是强化学习中一种经典的算法,其基本的思想是直接优化一个可迭代计算的 Q 函数. 本节将对比REINFORCE算法与Q-learning算法在求解该问题时的性能优劣.

选定场景3作为仿真环境,并将突发情况设置为

$T^*_{\mathrm{G1,sta_1}} = 5, 10, 15, \ldots, 50$ min十种不同的扰动. 其中$T^*_{\mathrm{G1,sta_1}} = 15$ min与$T^*_{\mathrm{G1,sta_1}} = 55$ min的Q-learning算法训练过程、REINFORCE算法训练过程以及最终的动态调度结果如图7所示.



(a) REINFORCE方法训练过程($T^*_{\mathrm{G1,sta_1}}$ = 15min)　(b) Q-learning方法训练过程($T^*_{\mathrm{G1,sta_1}}$ = 15min)　(c) 列车动态调度结果($T^*_{\mathrm{G1,sta_1}}$ = 15min)

(d) REINFORCE方法训练过程($T^*_{\mathrm{G1,sta_1}}$ = 45min)　(e) Q-learning方法训练过程($T^*_{\mathrm{G1,sta_1}}$ = 45min)　(f) 列车动态调度结果($T^*_{\mathrm{G1,sta_1}}$ = 45min)

图 7　REINFORCE与Q-Learning的结果对比

REINFORCE与Q-learning两种算法的计算结果时间对比如表9所示. 表9第3行object数据是两种方法对2.4节模型进行求解的结果,最终求解结果是一致的. 虽然两种方法都可以找到最优解,但是REINFORCE算法的计算时间远小于Q-learning算法.

表9　REINFORCE与Q-learning的计算时间对比　单位: s

	delay									
	5 min	10 min	15 min	20 min	25 min	30 min	35 min	40 min	45 min	50 min
REINFORCE	394.28	415.50	408.52	385.16	354.93	395.90	312.57	316.69	446.67	433.98
Q-learning	1 834.2	2 047.6	1 806.4	1 921.0	2 008.9	1 951.0	1 817.3	1 861.1	1 855.7	1 835.2
object/min	90	180	270	396	486	558	648	738	810	900

5　结　论

本文针对高铁延误的动态调度问题,设定延误列车在各站到发时间晚点总和最小为优化目标,构建了混合整数非线性规划模型,提出了基于策略梯度强化学习的动态调度方法. 最后对算法进行了仿真研究,并与Q-learning算法进行了比较. 结果表明,本文仿真环境的设计与算法是相适的,REINFORCE算法在求解本文模型上的性能明显优于Q-learning算法. 当然,本文只是对单线单向线路进行了研究,还有许多复杂的因素需要考虑,如双线双向、移动闭塞区间等因素,将在之后的工作中继续深入研究.

参考文献(References)

[1] Shakibayifar M, Sheikholeslami A, Corman F. A simulation-based optimization approach to reschedule train traffic in uncertain conditions during disruptions[J]. Scientia Iranica, 2018, 25(2): 646-662.

[2] Quaglietta E, Corman F, Goverde R M P. Impact of a stochastic and dynamic setting on the stability of railway dispatching solutions[C]. The 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013). Hague, 2013: 1035-1040.

[3] Butdee S. Simulation on high speed train with driving algorithm and rules for operation image control[J]. Procedia Manufacturing, 2019, 30: 575-580.

[4] Sovicka P, Pacha M, Rafajdus P, et al. Improved train simulation with speed control algorithm[J]. Transportation Research Procedia, 2019, 40: 1563-1570.

[5] Wang J H, Rakha H A. Longitudinal train dynamics model for a rail transit simulation system[J]. Transportation Research Part C: Emerging Technologies, 2018, 86: 111-123.

[6] Pellegrini P, Pesenti R, Rodriguez J. Efficient train re-routing and rescheduling: Valid inequalities and reformulation of RECIFE-MILP[J]. Transportation Research Part B: Methodological, 2019, 120: 33-48.

[7] 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[J]. 交通运输系统工程与信息, 2018, 18(4): 163-170.
(Jiang F, Ni S Q, Lv H X. A high-speed railway new-added train timetable partial adjustment model based on Lagrangian relaxation[J]. Journal of Transportation Systems Engineering and Information Technology, 2018, 18(4): 163-170.)

[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[J]. 铁道学报, 2016, 38(9): 1-8.
(Liao Z W, Miao J R, Meng L Y, et al. An optimization algorithm for double-track railway train timetabling based on Lagrangian relaxation[J]. Journal of the China Railway Society, 2016, 38(9): 1-8.)

[9] Zheng Y J. Emergency train scheduling on Chinese high-speed railways[J]. Transportation Science, 2018, 52(5): 1077-1091.

[10] 孟慧慧, 王长林. 基于双重启发式动态规划算法的列车运行调整研究[J]. 铁路计算机应用, 2014, 23(8): 1-4.
(Meng H H, Wang C L. Train operation regulation based on dual heuristic programming algorithm[J]. Railway Computer Application, 2014, 23(8): 1-4.)

[11] Qi J G, Yang L X, Gao Y, et al. Integrated multi-track station layout design and train scheduling models on railway corridors[J]. Transportation Research Part C: Emerging Technologies, 2016, 69: 91-119.

[12] Zhu Y Q, Goverde R M P. Railway timetable rescheduling with flexible stopping and flexible short-turning during disruptions[J]. Transportation Research Part B: Methodological, 2019, 123: 149-181.

[13] Niu H M, Tian X P, Zhou X S. Demand-driven train schedule synchronization for high-speed rail lines[J]. IEEE Transactions on Intelligent Transportation Systems, 2015, 16(5): 2642-2652.

[14] Xu X M, Li K P, Yang L X. Scheduling heterogeneous train traffic on double tracks with efficient dispatching rules[J]. Transportation Research Part B: Methodological, 2015, 78: 364-384.

[15] 林博, 俞胜平, 刘子源, 等. 基于改进粒子群算法的高铁列车动态调度[J]. 控制工程, 2021, 28(7): 1334-1341.
(Lin B, Yu S P, Liu Z Y, et al. High-speed train dynamic scheduling method based on improved particle swarm optimization algorithm[J]. Control Engineering of China, 2021, 28(7): 1334-1341.)

[16] Yu S P, Lin B, Zhang T, et al. Dynamic scheduling method of high-speed trains based on improved particle swarm optimization[C]. 2018 International Conference on Intelligent Rail Transportation (ICIRT). Singapore, 2018: 1-5.

[17] Šemrov D, Marsetič R, Žura M, et al. Reinforcement learning approach for train rescheduling on a single-track railway[J]. Transportation Research Part B: Methodological, 2016, 86: 250-267.

[18] Khadilkar H. A scalable reinforcement learning algorithm for scheduling railway lines[J]. IEEE Transactions on Intelligent Transportation Systems, 2019, 20(2): 727-736.

[19] Zhu Y Q, Wang H R, Goverde R M P. Reinforcement learning in railway timetable rescheduling[C]. The 23rd International Conference on Intelligent Transportation Systems (ITSC). Rhodes, 2020: 1-6.

[20] Ning L B, Li Y D, Zhou M, et al. A deep reinforcement learning approach to high-speed train timetable rescheduling under disturbances[C]. 2019 IEEE Intelligent Transportation Systems Conference (ITSC). Auckland, 2019: 3469-3474.

[21] Gong I, Oh S, Min Y H. Train scheduling with deep Q-network: A feasibility test[J]. Applied Sciences, 2020, 10(23): 8367-8381.

[22] 张琦, 陈峰, 张涛, 等. 高速铁路列车连带晚点的智能预测及特征识别[J]. 自动化学报, 2019, 45(12): 2251-2259.
(Zhang Q, Chen F, Zhang T, et al. Intelligent prediction and characteristic recognition for joint delay of high speed railway trains[J]. Acta Automatica Sinica, 2019, 45(12): 2251-2259.)

[23] Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators[J]. Neural Networks, 1989, 2(5): 359-366.

作者简介

俞胜平 (1976–), 男, 副教授, 博士, 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;

韩忻辰 (1996–), 男, 硕士生, 从事深度强化学习与调度理论及应用的研究, E-mail: hanxinchen@stumail.neu.edu.cn;

袁志明 (1980–), 男, 研究员, 博士, 从事行车指挥自动化、列车运行控制、智能调度和多列车协同控制等研究, E-mail: 13810696163@139.com;

崔东亮 (1976–), 男, 讲师, 博士, 从事高铁智能调度的研究, E-mail: cuidongliang@mail.neu.edu.cn.

(责任编辑: 闫 妍)