中国科学: 信息科学 2022 年 第 52 卷 第 5 期: 890–906
SCIENTIA SINICA Informationis
论文

基于强化学习的高速列车群运行调整方法

代学武[1*], 程丽娟[1], 崔东亮[1], 俞胜平[1], 袁志明[2], 应志鹏[2]

1. 东北大学流程工业综合自动化国家重点实验室, 沈阳 110819
2. 中国铁道科学研究院集团有限公司, 北京 100089
* 通信作者. E-mail: daixuewu@mail.neu.edu.cn

收稿日期: 2021–03–02; 修回日期: 2021–05–25; 接受日期: 2021–06–24; 网络出版日期: 2022–05–17

国家自然科学基金 (批准号: 61790574, 61773111, U1834211)、辽宁省自然科学基金 (批准号: 2020-MS-093)、国家铁路集团科技研究开发计划 (批准号: N2019G020) 和兴辽英才计划 (批准号: XLYC1808001) 资助项目

摘要 随着我国高速铁路建设成网, 行车密度不断提高, 在出现突发事件导致列车晚点时, 行车调度的复杂性和难度急剧增加, 如何动态调整列车群运行, 以减少晚点, 提高准点率是运行调整的核心. 本文提出了一种适用于突发事件下高速列车群运行调整的无模型强化学习方法. 首先将多个列车在多个车站和闭塞区间的运行调整建模为受约束的资源占用和配置的多阶段序贯决策过程, 提出了基于动态时空拓扑矩阵的车站和区间统一化建模方法. 针对高铁列车群时空关联强的特点, 首次提出了一种包含车辆位置、路网资源等时空分布信息的强化学习状态空间、动作空间和回报函数, 构建了有效的奖励反馈机制. 然后, 针对高铁运行系统搜索空间巨大的难点, 提出了启发式动作子空间自适应生成方法, 利用部分显式静态约束构建启发式规则减少搜索空间, 有效减少了无模型强化学习的试错次数, 提高了求解效率, 也保留了无模型通用性好的优点. 最后, 基于京广高铁实际案例的仿真分析表明, 在发生不同时空范围的大风限速, 导致多车延误的事件下, 所提出的算法均能较好收敛, 明显减少列车群内晚点传播, 与 MILP, ACO, FCFS 方法相比, 列车群的平均晚点时间可减少 2%∼20%.

关键词 强化学习, 时空拓扑矩阵, 列车运行调整, FCFS 算法, 优化

1 引言

高速列车在运行过程中由于气候、灾害、设备故障等突发事件导致实际运行轨迹偏离预定运行计划时, 需要对列车运行进行调整. 列车运行调整优化是指在列车运行延误和紊乱时, 动态地根据列车和路网的实际运行状态调整列车运行计划, 使得晚点列车能尽快地恢复按图运行或者尽可能地减少晚点, 从而减少晚点列车数量和受影响范围. 随着我国高铁建设成网, 至 2020 年底我国高铁运营里程已达 3.9×10^4 km, 占全球总里程 65% 以上. 日趋复杂的路网和不断提高的客流量、行车速度和行车密度,

引用格式: 代学武, 程丽娟, 崔东亮, 等. 基于强化学习的高速列车群运行调整方法. 中国科学: 信息科学, 2022, 52: 890–906, doi: 10.1360/SSI-2021-0073
Dai X W, Cheng L J, Cui D L, et al. Rescheduling of high-speed trains: a reinforcement learning approach (in Chinese). Sci Sin Inform, 2022, 52: 890–906, doi: 10.1360/SSI-2021-0073

给高铁列车运行调整提出了新的要求. 目前世界各国高铁均采用了调度集中系统 (centralized traffic control, CTC), 其中有代表性的如欧洲的 ETML, 日本的 COSMOS 系统等. 我国于 2003 年在列车调度指挥系统 (train operation dispatching command system, TDCS) 的基础上, 建设了新一代的分散自律调度集中系统. 这些 CTC 系统的应用改善了数据获取和调度指令下发的自动化程度, 但列车运行调整计划的制订还是由调度员借助计算机辅助绘图工具凭经验完成, 不仅劳动强度大, 且受人员经验和习惯的影响明显. 我国高铁线路繁忙、运输能力紧张、行车密度大、列车关联性强、跨线列车多、交路长、相互牵制面广, 因此突发事件下列车群运行调整难度极大. 利用计算机技术对列车群运行调整方案进行智能优化, 是智能高铁的重要发展方向.

列车运行调整问题是 NP 难问题 [1], 其求解方法如图 1 所示, 可以分为运筹学方法 [2]、仿真方法 [3] 和人工智能 [4] 方法三大类. 运筹学方法通常将列车运行调整过程建模为混合整数线性规划 (mixed integer linear programing, MILP) 模型 [5], 进而采用分支界定法 [6,7] 和动态规划及其各种改进方法求解, 如列生成 [8] 等. 也有学者将列车运行调整建模为车间作业调度问题 (job shop scheduling problem, JSSP), 采用析取图 (disjunctive graph, DG) [9] 或替代图 (alternative graph, AG) [10] 来求解, 其本质也是 MILP 优化问题. 运筹学方法有较完善的理论基础, 但是对于高铁这样庞大复杂的动态时变、关联强、安全和设备约束多的系统, 难以建立准确的过程模型, 通常会作较多假设和简化, 所得解的可行性还需进一步验证. 且随着问题规模的扩大、约束条件的增多, 运筹学方法的时效性和可行性等缺点更加明显 [11]. 第 2 类方法是基于仿真的方法, 所获得的解有更好的可行性, 如 zhou 等 [3] 基于离散事件动态系统 (discrete event dynamic systems, DEDS) 理论和滚动优化策略提出了列车运行计划与调整的计算机实现方法, Xu 等 [12] 基于列车运行的仿真模型研究了单线铁路的运行图调整方法, 但计算机仿真的方法存在计算量太大, 实时性较差的问题.

另一方面, 作为一种高效的近似求解算法, 一系列基于人工智能的启发式优化算法被应用于列车运行调整, 如蚁群优化 [13]、粒子群 [14]、遗传算法 [15] 和禁忌搜索 [16] 等. 与运筹学方法相比, 这类基于群体智能的优化方法在目标函数优化的迭代搜索过程中, 更好地集成了领域知识和约束条件,能够较快地求得一个近似的全局满意解, 适合于约束多、模型不精确的优化问题 [17]. 但由于采取群体优化方式, 目标函数的评估计算量大, 导致求解大规模问题时效率低和实时性差. 大部分基于运筹学和智能优化的列车运行调整方法 (不包括 DP), 其基本思路都是将列车调整问题视作一个单阶段决策的静态过程, 对所有决策变量作为一个整体进行搜索, 以最小化目标函数. 值得指出的是, 列车运行调整是一个时空约束多的多阶段序列决策 (sequential decision making, SDP) 动态过程, 调度方案需要随着路网运行状态的改变而动态改变.

与基于群体智能的优化算法不同, 强化学习是一种适合于快速求解复杂大系统序列决策的优化方法, 已在游戏、机器人路径搜索、生产调度等领域取得了成功应用. 强化学习方法通常分为有模型的强化学习和无模型强化学习 [18], 两者的区别在于前者需要环境模型 (状态转移矩阵) 已知, 可利用该模型进行学习, 而后者无需提前获知准确的模型, 智能体直接与环境进行实时交互学习最优策略, 对于复杂的应用场景来说后者具有更好的通用性. 近年来部分学者也开始利用强化学习策略求解列车动态调度问题, 但总体而言, 该研究在国内外都处于起步阶段 [19~21], 相关报道还很少. 文献 [21] 首次将 Q 学习应用到单线普速铁路调度, 求解质量超过了传统启发式算法. 为克服文献 [21] 中状态空间随问题规模急剧增加的缺点, 文献 [20] 采用列车前后若干个区间的占用状态作为状态变量, 不再包含列车的位置信息, 具有较好的扩展性. 但假定列车具有相同行为特征, 通过学习单个列车的调度策略, 多个列车复用同一 Q 值表, 虽能解决多车调度和死锁, 但无法处理相同等级的列车越行问题. 文献 [19] 采用了 deep Q-network 算法, 但状态空间是全局时刻表, 难以避免维度灾难. 基于机器学习的方法还有神

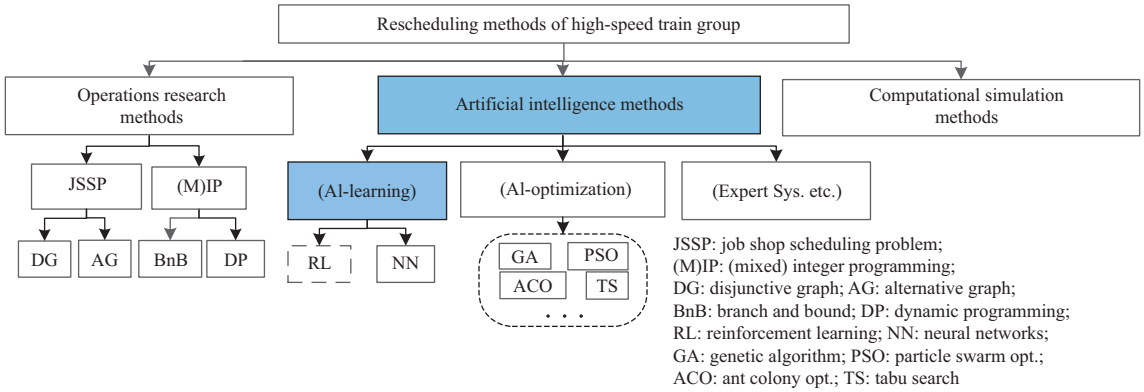

图 1 (网络版彩图) 列车群运行调整问题的求解方法

Figure 1 (Color online) Research methods for rescheduling high-speed trains

经网络方法 [4], 但在列车调度中应用相对较少, 通常与其他优化方法联合运用.

本文针对传统调度优化算法中采用单阶段静态决策和直接对决策变量整体进行搜索的局限, 构建了高铁列车群运行调整的 SDP 模型, 考虑列车运行过程的动态特性和随机性, 以及部分隐式约束难以表达的特点, 本文提出了一种新型的基于无模型强化学习的列车群运行调整体系架构和求解算法, 有效克服了高铁调度的马尔科夫 (Markov) 决策过程的状态转移函数难以获得以及约束条件复杂等难题. 与现有强化学习列车调度 [19~21] 相比, 本文针对列车群多车调度, 不是单车的学习, 在状态变量定义上采取了关联区间占用和列车位置相结合的方式, 避免了列车群调度的 “维度灾难” 问题, 采用了启发式动作子空间自适应生成方法提高性能, 主要贡献如下:

(1) 从时间空间两个角度, 将列车群运行调整过程建模为资源的占用、释放和再分配的序贯决策过程, 提出了基于 “动态时空拓扑矩阵” 的建模方法, 对车站和区间统一化建模, 通用性更好;

(2) 考虑列车群时空关联强的特性, 首次提出了一种包含多个关联列车的位置和路网资源时空占用信息的强化学习状态空间和动作空间, 构建了有效的回报和奖励反馈机制, 确保算法收敛;

(3) 针对搜索空间巨大的问题, 提出了启发式动作子空间自适应生成方法, 利用部分显式静态约束构建启发规则, 减少了无模型强化学习的试错次数, 提高求解效率的同时, 保留了无模型通用性好的优点, 具有将高铁调度中复杂的约束关系 (如联锁等) 在所构建的环境模型中进行表征, 并从环境中学习到这些约束的能力.

本文结构如下: 第 2 节构建了列车群运行调整的序贯决策模型及其强化学习体系框架; 第 3 节给出了列车群运行调整的状态向量、动作空间和奖励函数, 以及动作空间自适应生成方法; 最后以京广线高密度行车区段的实际案例进行验证, 并与常用的调度算法 MILP, ACO (ant colony optimization), FCFS (first-come first-service) 进行对比, 表明了本文所提方法的有效性.

2 问题描述

高速列车群在路网上的运行是一个实时动态变化的过程, 且约束多, 列车之间耦合关系复杂. 本文从时空资源的占用角度, 构建面向强化学习的列车运行调整的序贯决策模型.

2.1 列车群运行调整的序贯决策建模

高铁运行是在由若干车站和区间连接而成的铁路轨道路网上的一组高速列车按照预定的运行计

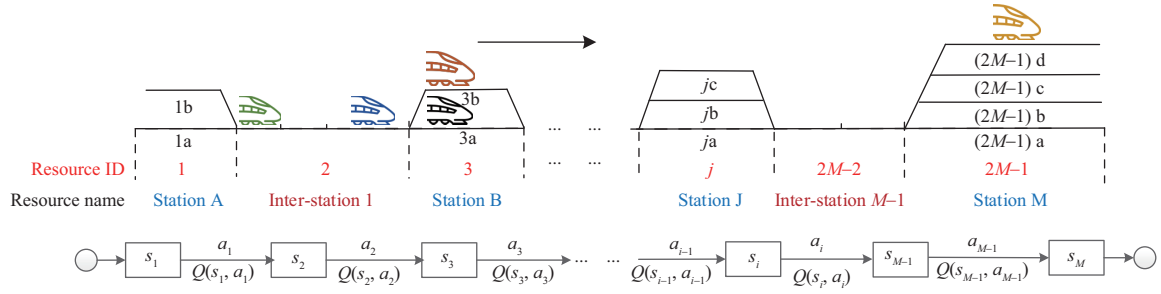

图 2 (网络版彩图) 高铁调度区段模型

Figure 2 (Color online) High-speed rail dispatching section model

划和一系列安全约束等规则行车的过程. 不失一般性, 本文将车站和区间统一视作资源块. 因此列车在区间运行或者在车站通过或者停车等作业时, 可以视作列车对资源块的占用和释放过程. 如图 2 所示 2 个车站和 3 个区间组成某调度区段下行线路布局, 当列车在区间资源块 2 上完成作业时, 如果后继资源块 3a 和 3b 都被占用, 此时列车应当继续停留在资源块 2 上直到资源块 3a 或者 3b 被释放. 值得指出的是, 车站可被视作是一个具有多个并行股道的资源块, 在符合联锁和进路安排等约束条件下, 可以同时被多个列车占用. 与此类似, 区间资源块在遵守信号系统行车间隔等安全约束条件下, 可以允许多个列车在同一区间保持安全距离运行, 如图 2 中资源块 2 中的前后运行的两个列车.

综上所述, 列车运行调整问题实质上是有限资源的供给、消耗与再分配问题. 因此, 本文将列车群的动态运行建模为资源、约束和事件在时间和空间上分布及其动态变化的过程, 提出了 “动态时空拓扑矩阵” 这种统一的形式进行描述和规范化表达. 首先对列车在路网中的运行过程提出部分假设, 描述如下:

(1) 连通性. 列车可以从一个资源中的任意轨道移动到下一个邻接资源中的任意可用轨道.

(2) 当列车从当前资源离开进入下一个资源时, 由于列车运行速度快, 切换时间很短, 所用时间可近似为 0.

设某调度区段的下行 (或上行) 线路包含 M 个车站和 $M-1$ 个区间分别编号为 $m \in \mathbb{M} = \{1,2,3,\ldots,M\}$ 和 $b \in \mathbb{B} = \{1,2,3,\ldots,M-1\}$. 为了方便描述, 将车站和区间均统一建模为资源块, 则共有 $R = 2\times M-1$ 个资源块编号为 $j \in \mathbb{R} = \mathbb{B}\cup\mathbb{M} = \{1,2,3,\ldots,R\}$. 车站作为特殊的资源块通常有多个股道, 其容量为 C_j, 对股道分别用字母 $\{\mathrm{a},\mathrm{b},\mathrm{c},\ldots\}$ 编号, 如图 2 中 Station B, 其资源块编号为 3, 有两个股道分别编号为 3a, 3b. 运行调整计划涉及 N 辆列车, 列车按照在调度区段始发站的计划发车顺序进行编号为 $i \in \mathbb{N} = \{1,2,3,\ldots,N\}$.

本文中, 描述列车群运行时空动态变化过程的 “动态时空拓扑矩阵” 可以用三元组 (t,j,r) 来定义, 记作 $r(j,t)$, 其中 t 是资源状态发生切换的时间, 为离散时间域, $j \in \mathbb{R}$ 是资源块编号, r 为资源的多维属性, 可以用来描述资源状态、列车最小运行时间等. 本文仅采用其中的占用状态属性, $r(j,t) \in \{0,1,\ldots,C_j\}$, 将 j 和 t 作为二维矩阵下标. 如图 3 是一个动态时空矩阵的示例, 纵坐标表示路网资源编号, 横坐标为时间. 不同颜色数字表示不同列车对资源占用和释放的过程, $r(j,t)=1$ 表示资源空闲, $r(j,t)=0$ 表示资源 j 被一个列车占用, 以此类推, $r(j,t)=C_j$ 则表示该资源已全部被占用. 图 3 中左侧分别展示了 $t=1$, $t=4$, $t=7$ 时刻下路网资源的占用情况, 分别对应图中右侧矩阵中的第 1, 4, 7 列, 矩阵中不同颜色的 0 表示资源块 j 被对应颜色列车占用. 通过采用时空拓扑矩阵表达事件的注入、资源的占用和约束等的方式可以很容易转换成各种事件序列. 把列车 i 对资源 j 占用表示为一个活动

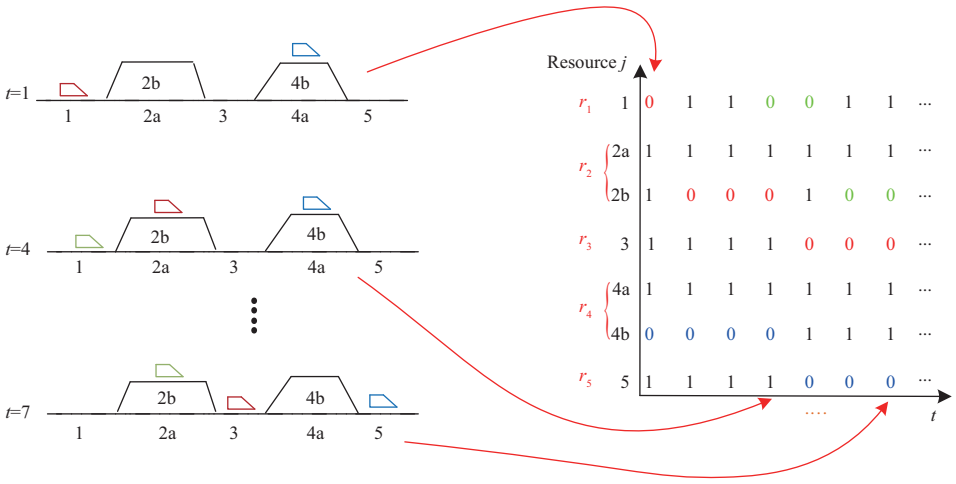

图 3　(网络版彩图) 资源占用矩阵

Figure 3　(Color online) Resource occupation matrix

$e_{i,j}$, 则列车 i 的运行过程可以看成由一系列有序的活动 E 组成, $E=\{e_{i,j_0},e_{i,j_1},\ldots,e_{i,j_k},e_{i,j_k+1},\ldots\}$, 这里 $j_k,j_{k+1}\in\mathbb{R}$ 表示列车运行径路上前后相邻的两个资源块, j_0 表示列车 i 的始发站, (e_{i,j_k},e_{i,j_k+1}) 为相邻活动.

2.2　列车群运行调整目标和约束

路网中运行的每个列车都有各自的运行计划, 可以用列车在铁路区间运行及在车站到发或通过时刻来描述, 定义如下:

定义1 (到站时间 $s_{i,j}$)　$s_{i,j}$ 用来表示活动 $e_{i,j}$ 的开始时间, 即列车 i 对资源 j 开始占用的时间. 当资源 j 为车站时, $s_{i,j}$ 就是列车 i 在车站 j 的到达时间.

定义2 (发车时间 $d_{i,j}$)　$d_{i,j}$ 用来表示活动 $e_{i,j+1}$ 的开始时间, 即列车 i 对资源 $j+1$ 开始占用的时间. 当资源 j 为车站时, $d_{i,j}$ 就是列车 i 在车站 j 的发车时间.

列车的运行计划可以图形化表示为时间和空间上的运行线, 称作运行图. 基本运行图是理想情况下所期望的列车运行计划. 列车实际运行过程中受突发事件影响, 列车的实际到发时间偏离了基本运行图, 造成列车晚点甚至运行秩序紊乱. 列车运行调整的目的是使晚点列车尽可能恢复按图运行或者尽可能减少与图定时间的偏离. 常用的列车运行调整优化的目标有总晚点时间、晚点列车数量、晚点增量等 [10,22]. 不失一般性, 本文采用总晚点时间作为优化目标

$$\begin{aligned}&\min J=J_a+J_d,\ \text{where}\ J_d=\sum_{i\in\mathbb{N}}\sum_{j\in\mathbb{M}}|d_{i,j}-d^*_{i,j}|\\&\text{and}\ J_a=\sum_{i\in\mathbb{N}}\sum_{j\in\mathbb{M}}\begin{cases}0, & -\delta<s_{i,j}-s^*_{i,j}<0,\\ |s_{i,j}-s^*_{i,j}|, & s_{i,j}-s^*_{i,j}>0\ \text{or}\ s_{i,j}-s^*_{i,j}<-\delta,\end{cases}\end{aligned}\tag{1}$$

其中, J_a 为到站总晚点, 值得注意的是, 对于列车早到 δmin 之内的情况视为 0, 本文设定 $\delta=2$; J_d 为发车总晚点; $s^*_{i,j}$ 表示列车 i 在车站 j 的图定到达时间; $d^*_{i,j}$ 表示列车 i 在车站 j 的图定发车时间. 列车在运行过程中, 还受到设备和安全运行等约束, 常见的约束包括以下 6 种.

(1) 车站容量约束

$$\sum_i x_{i,j,t} \leqslant C_j, \quad j \in \mathbb{R},\ i \in \mathbb{N}, \tag{2}$$

其中, $x_{i,j,t}$ 表示列车 i 对资源 j 在 t 时刻是否占用, $x_{i,j,t}=1$ 表示当前资源被占用, $x_{i,j,t}=0$ 表示未占用. 在任意时刻 t 占用资源 j 的列车的总数不能超过该资源的最大容量 C_j.

(2) 车站停站时间约束

$$d_{i,j} = s_{i,j} + a_{i,j}, \quad j \in \mathbb{R},\ i \in \mathbb{N}, \tag{3}$$

$$a_{i,j} \geqslant A_{i,j}, \quad j \in \mathbb{R},\ i \in \mathbb{N}, \tag{4}$$

其中, $a_{i,j}$ 表示列车 i 在资源 j 上实际停站时间; $A_{i,j}$ 表示列车 i 在资源 j 上的图定停车作业时间. 该约束反映了列车停站时间不应缩短, 以保证有足够时间完成旅客乘降作业.

(3) 区间最小运行时间约束

$$s_{i,j'} = d_{i,j} + t_{i,j}, \quad j, j' \in \mathbb{R},\ i \in \mathbb{N}, \tag{5}$$

$$t_{i,j} \geqslant T_{i,j}, \quad j \in \mathbb{R},\ i \in \mathbb{N}, \tag{6}$$

其中, $t_{i,j}$ 表示列车 i 在资源 j 上实际运行时间; $T_{i,j}$ 表示列车 i 在资源 j 上最小区间运行时间.

(4) 发车时间约束

$$d_{i,j} \geqslant d^*_{i,j}, \quad j \in \mathbb{M},\ i \in \mathbb{N}. \tag{7}$$

在实际运行过程中, 列车的实际发车时间一般不允许小于图定计划发车时间.

(5) 进站或者离站时间间隔约束

$$s_{i,j} - s_{i',j} \geqslant g_j, \quad j \in \mathbb{M},\ i, i' \in \mathbb{N}, \tag{8}$$

$$d_{i,j} - d_{i',j} \geqslant g_j, \quad j \in \mathbb{M},\ i, i' \in \mathbb{N}, \tag{9}$$

其中, g_j 表示列车 i 和相邻列车 i' 离站与进站最小时间间隔. 即相邻列车的离站/进站时间间隔应不小于 g_j.

(6) 同方向不同时到发和发到时间间隔约束

$$s_{i,j} - d_{i',j} \geqslant \tau_{sd}, \quad j \in \mathbb{M},\ i, i' \in \mathbb{N}, \tag{10}$$

$$d_{i,j} - s_{i',j} \geqslant \tau_{ds}, \quad j \in \mathbb{M},\ i, i' \in \mathbb{N}, \tag{11}$$

其中, τ_{sd} 表示列车 i 和相邻列车 i' 之间的同方向列车不同时到发最小时间间隔, τ_{ds} 表示列车 i 和相邻列车 i' 之间的同方向列车不同时发到最小时间间隔. 即当相邻列车之间的不同时到发时间间隔小于 τ_{sd} 时或者当相邻列车之间的不同时发到时间间隔小于 τ_{ds} 时, 将发生冲突.

3 列车群运行调整强化学习

本文提出的用于列车群运行动态调整的强化学习, 主要采用列车运行仿真环境与 Q 值学习相结合的方式, 如图 4 所示, 通过算法与环境大量模拟和离线训练获得较好的调度策略, 训练完成的调度算法也具有较好的实时性.

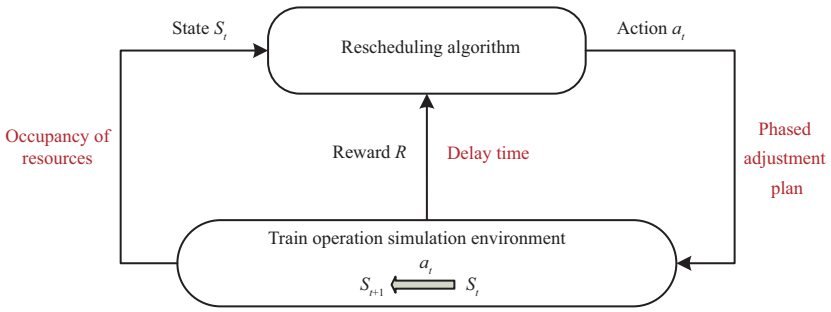

图 4　(网络版彩图) 列车群动态调整的强化学习体系框架

Figure 4　(Color online) Reinforcement learning system framework for dynamic adjustment of train operation

其基本思路是调度算法作为智能体通过序列决策的方式进行求解. 首先从目标环境 (即列车运行过程) 中获取当前路网状态 S_t, 并根据某种策略, 从所有可行的动作中选择一个动作 a_t (即列车的到发), 在该动作的驱动下列车运行进入新的状态 S_{t+1} 并给出一个可反映新状态优劣的奖励值 r_{t+1}. 然后智能体根据新的状态和奖励值再次进行决策选择动作, 以此往复, 直到所有列车到达调度区段的最后一站 (称作交出站). 通过智能体和列车运行环境之间不断交互, 智能体通过更新 Q 值逐渐地学习得到最佳的值函数, 使得长期回报最大化, 最终得到较好的调度策略.

本文所提算法的关键是将列车运行调整的调度优化问题映射为强化学习中根据列车运行状态选择最优动作的问题. 由于列车运行过程可以看作一系列的状态切换过程, 下一个时刻的状态取决于当前时刻列车对资源的占用情况和所选择的动作, 因此本文从时间和空间两个角度定义 Q 学习所需的状态向量、动作空间和奖励函数.

3.1　状态向量

路网状态的评价不仅和列车的位置、股道资源的占用等相关, 还取决于该状态所处的时间. 因此状态向量不仅要能够反映列车运行过程中路网资源的状态和变化特征, 还需要反映出时间维度的特征. 列车运行调整问题与很多因素相关, 主要包括: 路网条件如资源块的顺序、名称, 以及股道的数量等; 移动设备状态如列车数量、编号、速度和计划时刻表; 以及一些约束条件等. 从理论上可以将调度时间范围 $[t_1, t_2]$ 进行离散化处理得到集合 $\mathbb{T} = \{t_1, t_1+1, t_1+2, t, \ldots, t_2\}$. 针对高速列车调度问题的特点, 从时间和空间两个角度定义 $(2 \times N + 1)$ 维向量为系统状态 S_t:

$$S_t = [h_{1,t}, h_{2,t}, h_{i,t}, \ldots, h_{n,t}, O_{c_{1,t},t}, \ldots, O_{c_{i,t},t}, \ldots, O_{c_{n,t},t}, t] \quad h_{i,t} \in \mathbb{J},\ c_{i,t} \in \mathbb{J}^+,\ \text{and}\ t \in \mathbb{T}, \tag{12}$$

其中, $h_{i,t} \in \mathbb{R}$ 为列车 i 在时刻 t 的相对位置, 即所在资源块编号. 下标 $c_{i,t}$ 为列车 i 在前进方向的下一个资源块编号, 单线正向运行等简单场景下 $c_{i,t} = h_{i,t} + 1$, 但复杂路网下 $c_{i,t}$ 和 $h_{i,t}$ 可能存在跳变. 因此, 用 $\mathbb{J}$ 表示当前时刻所有列车占用资源块集合, $\mathbb{J}^+$ 为列车前进方向的下一个资源块的集合.

定义3 (资源状态 O_j)　为简化符号, 令 $j = c_{i,t}$, 则 O_j 代表是否允许列车从当前资源块 $h_{i,t}$ 前进占用下一资源块 j. 该状态主要反映了下一资源块 j 的占用状态, 以及最小停站时间、发车时间等约束. 若 t 时刻占用资源块 j 的列车总数 $\sum_i x_{i,j,t}$ 小于该资源容量 C_j, 并且列车 i 在该时刻已满足最小停站时间等运行约束, 则 O_j=1, 否则 O_j=0.

这样定义的好处不只是根据列车当前位置进行决策, 还考虑了临近资源块的拥挤程度, 用资源的占用率进行衡量, 使得作出的决策更真实有效.

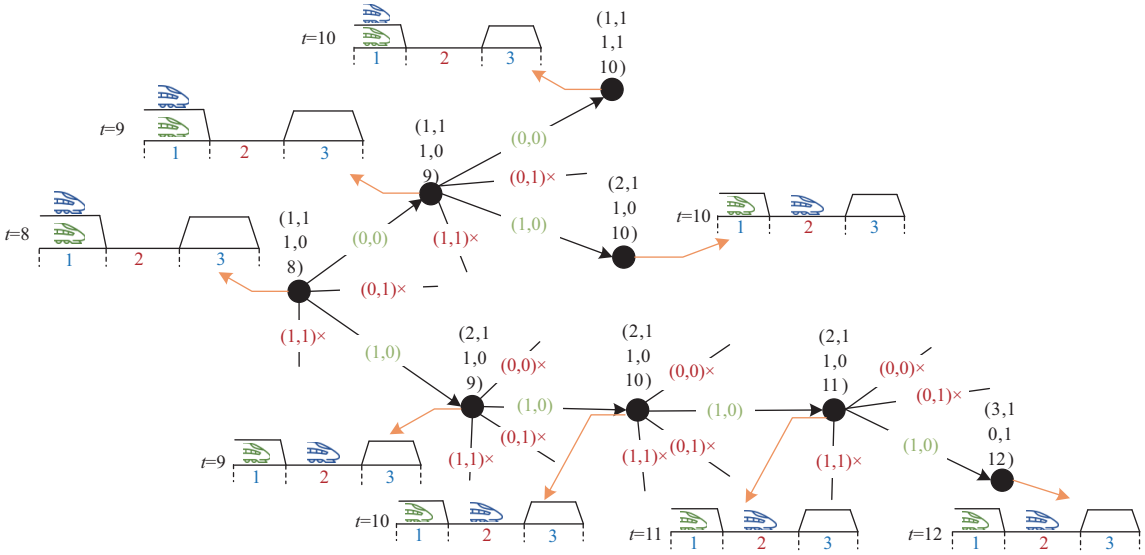

图 5 (网络版彩图) 两车调度中当前状态和可能的相应动作的 “状态 – 动作” 树

Figure 5 (Color online) The “state-action” tree of the current state and possible corresponding actions in a two-train scheduling scenario

3.2 动作空间

列车运行调整问题是当列车发生晚点时, 需要根据实时的列车位置和路网资源占用状态选择适当的动作, 决定当前时刻哪些列车应发车进入下一个资源块, 哪些列车应继续保持在当前资源上. 因此对于 N 趟列车的情况, 动作向量可定义为一个 N 维向量 $\boldsymbol{a} = [a_1, a_2, a_i, \ldots, a_n]$, 其中当 $a_i = 1$ 表示列车 i 进入下一个资源块上, 当 $a_i = 0$ 时表示列车 i 将继续停留在当前资源块上, 列车将在下一个决策步进行重新决策直到移动至下一个资源上. 值得指出的是, N 维的动作空间中所有可能出现的动作数的大小为 2^N, 随着问题规模的扩大, 其大小呈指数增长.

在序列动作 $\{a_t|t = 1, 2, \ldots\}$ 的作用下, 列车运行不断从一个状态切换到另一个状态, 以此往复, 直到到达目标状态. 这样一个过程可以用 “状态 – 动作” 树来描述. 当列车数量较多时可行的动作空间会非常庞大, 导致搜索时间急剧增加. 由于 Q 学习是一种试错式的学习, 在学习的开始阶段智能体未从交互数据中学到有用的知识. 在不了解环境和列车运行过程的约束条件时, 需要在维度较大的动作空间随机探索进行大量试错, 逐步学习到列车运行过程中的各种约束条件, 造成大量的无效的学习使得学习时间大大延长甚至难以收敛.

3.3 动作子空间自适应生成

为了提高算法收敛速度, 本文结合列车调度的特点考虑列车运行过程中的两类典型约束条件: (1) 静态可提前获知的, 如式 (2)~(11) 的到发间隔约束等; (2) 在运行过程中动态变化的, 或难以提取、较难形式化描述的, 如通过列车和办理乘降业务的列车之间的发车间隔. 这种约束与 (10) 和 (11) 不一样, 尽管此约束条件尚不完善, 也应在算法中予以考虑, 以避免初始阶段对大量无效动作的学习, 从而加快 Q 学习的收敛速度. 因此, 在算法中根据当前状态 S_t 以及约束条件生成合法的动作子空间, 或者将不符合目标函数优化的动作从动作空间中剔除, 有效减少了可行动作的集合规模, 智能体在当前状态 S_t 下只需在可行的动作中选择.

图 5 以两辆列车运行过程的 “状态 – 动作” 树为例, 图中黑色实心圆点表示状态, 两个状态的连接线上的红色数字向量和绿色数字向量表示动作, 在该动作的驱动下从上一个状态切换到下一个状态. 图中第 1 个黑色实心点被称为根节点, 其状态向量为 $S_0 = [1,1,1,0,8]$, 其中前两个元素表示在第 8 min 时两辆列车均位于编号为 1 的始发站; 元素 $[1,0]$ 表示在列车行驶方向上的下一个资源块的拥挤程度以及是否满足约束条件公式 (2)∼(11). 在无约束条件的限制下, 两辆列车对应的所有可能的动作为 $\{[0,0],[0,1],[1,0],[1,1]\}$. 但是由于受到约束条件公式 (2)∼(11) 的影响, 当列车运行不满足约束条件时, 或者当前列车所在位置的下一资源块处于拥挤程度较高的状态时, 例如 S_0 中第 4 个元素 $O_{c_{2,8},8} = 0$ 表示列车 2 下一个时刻不允许移动至下一个资源块 $c_{2,8}$ (即图中的编号为 2 的资源块), 此时 $a_2 = 1$ 是非法的, 因此动作 $(0,1)$ 和 $(1,1)$ 是不允许执行的操作, 如图 5 中红色叉号 (×), 智能体只需在可行的动作集内选择动作 (即图中绿色向量) 即可, 后续的状态按照以上规则类推. 这种设计的好处在于智能体在探索过程中只需在可行的非优动作集中进行选择, 减少对非法动作集的探索, 从而加快求解速度.

3.4 目标函数与奖励函数

列车晚点时长是列车运行的关键性能指标, 因此本文采用所有列车在所有车站总晚点时间最小作为目标函数 J, 见式 (1). 在 Q 学习求解过程中, 将延时奖励函数设置为与列车总晚点时间成反比, 记作 $r_T = K/J$, 其中 T 表示回合结束, K 是一个常量, 可以由用户自定义. 从奖励函数定义中可以看到列车总晚点 J 越大, 则 r_T 越小, 反之越大. 当列车的总晚点时间越大, 环境反馈的奖励函数值越小.

除了延时奖励函数, 在 Q 学习过程中每次状态发生转移时, 设计了即时奖励函数:

$$r_t = \begin{cases} -1, & \text{if } |d_{i,j} - s_{i,j}| > D, \\ 0, & \text{otherwise}, \end{cases} \tag{13}$$

其中, D 为用户自定义的常量. 为了避免出现大范围的停车以及发生超出调度时间域范围内还有列车没有到达终点站的现象, 本文考虑设置即时奖励值 $r_t = -1$, 在其他情况下即时奖励值 $r_t = 0$.

4 Q 学习求解算法设计

在定义了列车调度的状态、动作和奖励函数之后, 接下来的关键是合理地设计强化学习的训练算法, 对列车调度问题进行高效求解. Q 学习算法的核心是值函数 Q 的迭代学习, 值函数 Q 反映的是某个时刻 t 对应状态为 S_t 下衡量选择某个动作 a_t 的好坏程度. 在状态 S_t 下智能体选择动作 a_t 时, 根据该状态 – 动作对的 Q 值的大小选择动作, 使得智能体获得的累积奖励值最大化. 运用 Q 学习成功进行列车调度优化的关键在于是否能从大量数据中学习到能准确反映高速列车调度过程特征的值函数. 其中值函数更新公式为

$$Q(S_t,a_t) = (1-\alpha)Q(S_t,a_t) + \alpha\left(r + \gamma \max_{a_{t+1}\in A} Q(S_{t+1},a_{t+1})\right), \tag{14}$$

其中, $Q(S_t,a_t)$ 表示在当前时刻 t 的状态 S_t 采取动作 a_t 的效用函数, 其中 α 为步长因子, 又被称为学习率, 取值范围为 $\alpha \in (0,1]$, 需要根据实际问题特征选取. 当 α 越大表示当前奖励值对 $Q(S_t,a_t)$ 影响越大, 收敛速度越慢. $\gamma \in [0,1]$ 表示折扣因子, $\gamma \to 0$ 说明智能体最大化当前的奖励值, $\gamma \to 1$ 说明智能体对未来的奖励值更加注重. r 为在当前状态 S_t 下采取动作 a_t 得到奖励值.

考虑到高铁调度问题的动作空间维度随问题规模成指数级增加导致其状态空间非常巨大, 在有限的时间内不可能探索完所有动作空间, 并且容易陷入局部最优解, 反映在学习过程上就是 Q 值函数难以收敛. 为提高收敛速度, 在学习初期, 尽量基于约束规则和随机选择进行高效探索, 避免陷入局部最优解, 随着经验的增多, 中后期采用基于既有解路径来选择动作的思路, 逐步减少探索. 因此, 本文采用了 ε-greedy 的随机探索算法来平衡求解的实时性和解的全局性.

$$\pi(a|s)=\begin{cases}1-\varepsilon+\varepsilon/|A(s)|, & \text{if } a=\operatorname{argmax}(Q(s,A(s))),\\ \varepsilon/|A(s)|, & \text{if } a\neq\operatorname{argmax}(Q(s,A(s))),\end{cases}\tag{15}$$

其中, $\pi(a|s)$ 表示在该状态 s 下选择动作 a 的概率. 学习过程中产生一个随机数 $\text{rand}\in(0,1)$, 如果随机数小于 ε, 智能体在当前所有可选的动作集中等概率地随机选择动作, 反之选择值函数最大的动作值. 针对高铁调度过程, ε 值设置为

$$\varepsilon=\frac{0.8}{1+\mathrm{e}^{\frac{10\times(i-0.6\times N_{\text{itr}})}{N_{\text{itr}}}}},\tag{16}$$

其中, i 表示当前迭代次数, N_{itr} 表示学习过程最大的迭代次数. 这样设计的好处是能够平衡探索和利用两个过程, ε 值随着迭代次数的增加逐渐减小至 0, 表示 Q 学习算法学习求解过程中前期主要进行随机搜索动作, 随着次数增加后期主要选取 Q 值最大的动作. 根据上述分析得到列车运行调整优化的 Q 学习算法步骤如算法 1 所示.

Algorithm 1 Q-learning algorithm for rescheduling high-speed trains

Input: Initialize $Q(s,a)$ arbitrarily;
Output: Train rescheduling strategy;
for $i=1$ to N_{itr} **do**
  Initialize $S_t=S_0$;
  while $S_t\neq$ terminal stations S_{end} **do**
    Generate the feasible action space A according to the constraints (2)∼(11) as Subsection 3.2;
    Update ε (16) and choose an action a_t from A as (15);
    Take action a_t, obtain next state S_{t+1} and reward r_t from the envrionment;
    Update Q values as (14);
    S_t=S_{t+1};
  end while
end for
return Train rescheduling strategy.

5 性能验证与分析

为了验证本文所提方法的有效性, 本节基于行车密度较大的京广高铁 “武汉 – 郑州东” 区段的实际案例, 设置了 3 种代表性的典型运行场景. 通过仿真验证, 并与常用的 MILP, ACO [23], FCFS [7] 算法进行对比, 表明了本文所提强化学习调度方法的有效性. 仿真系统配置如下: 台式计算机 Intel Core i7-4790CPU@3.60 GH, 12 GB 内存, 基于 MATLAB2019a 搭建强化学习算法, MILP 算法采用了 Gurobi9.1.1 求解, ACO 算法采用 MATLAB2019a 求解.

表 1　所选调度区段路网参数

Table 1　Road network parameters of selected dispatching section

Station j	Location l_j (km)	Capacity C_j	Minimum running time (min)
1	0	5	10
2	81	3	11
3	136	3	10
4	201	2	20
5	297	3	13
6	361	4	21
7	473	3	–

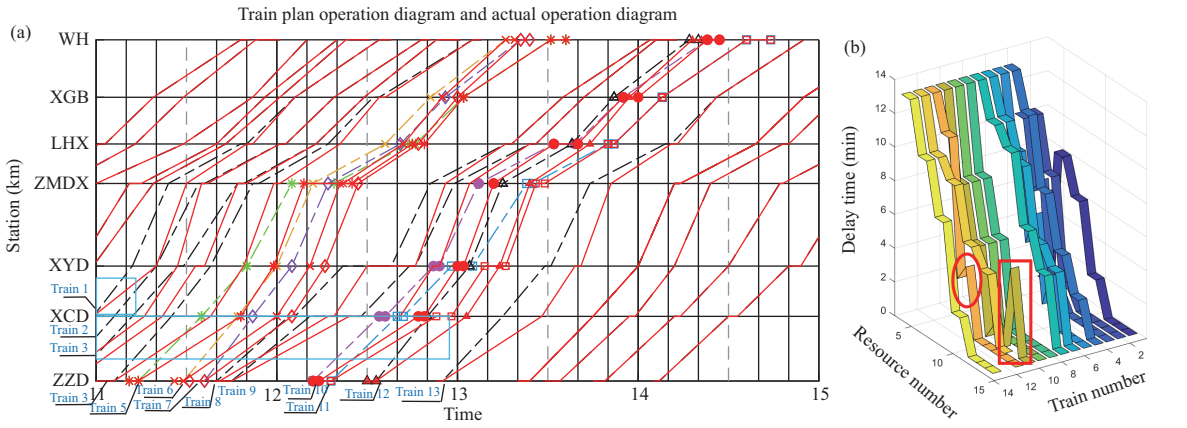

图 6　(网络版彩图) 案例 1 验证结果

Figure 6　(Color online) Results in case 1. (a) Planned train lines (dotted line) and rescheduled lines by the proposed Q-learning algorithm (solid line); (b) changes of each train's delays at every station

5.1　案例分析

以京广线 "武汉 – 郑州东" 区段 2019 年 3 月某日 11~15 时的运行场景为例, 该调度区段由 7 个车站组成, 依次编号为车站 1~7, 涉及 13 辆高速列车, 所选调度区段的路网参数见表 1, 其中以车站 1 作为 0 km 起点, 列出了车站 j 相对应的里程标 l_j 和车站股道容量 C_j, 以及从车站 j 到车站 $j+1$ 区间最小运行时间. 其他参数设置为 $g_j = \tau_{sd} = \tau_{ds} = W = 3$, $K = 6000$, Q 值初始化为 0, 学习率 $\alpha = 0.2$, 折扣因子 $\gamma = 0.9$. 最大迭代次数 $N_{\text{itr}} = 600$, ε-greedy 算法中 ε 根据式 (12) 从 0.8 逐渐减小至 0.

场景 1 (11:30 到 13:00 在车站 1 和 2 区间有大风临时限速). 限速区段的时空范围如图 6 蓝色方框所示, 虚线为基本图中的列车正点运行线. 受临时限速影响导致多辆列车晚点, 如列车 2 和 3 均晚点 10 min 到达车站 2, 后续列车从车站 1 正点发车, 但均会晚点 13 min 到达车站 2, 该列车群的初始晚点总和为 158 min.

针对该运行场景所构建的时空拓扑矩阵 $r(j,t)$, $j \in \mathbb{R}$, $t \in [11:00, 15:00]$, 如图 7 所示. 图 7(a) 是无临时限速时该列车群正常运行的时空资源的占用和释放分布, 不同的颜色代表不同列车, 总体来看, 行车比较密集, 区间资源占用率高 (如 Track2, 4, 6 等), 车站资源占用时间段相对都较短, 但繁忙时段接发列车密度高 (如车站 4 对应的 track4b 在 13:00 到 13:20 期间), 正确反映了高铁实际运行过程和状态.

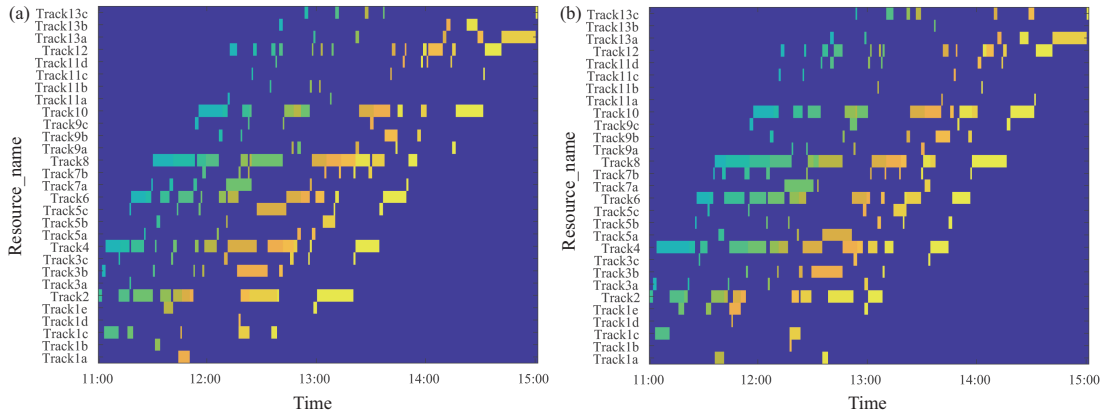

图 7　(网络版彩图) 场景 1 的时空拓扑矩阵

Figure 7　(Color online) Resource utilization chart by the proposed spatio-temporal topology matrix in scenario 1. (a) Resource occupation before disturbance occurs; (b) resource occupation after disturbance occurs

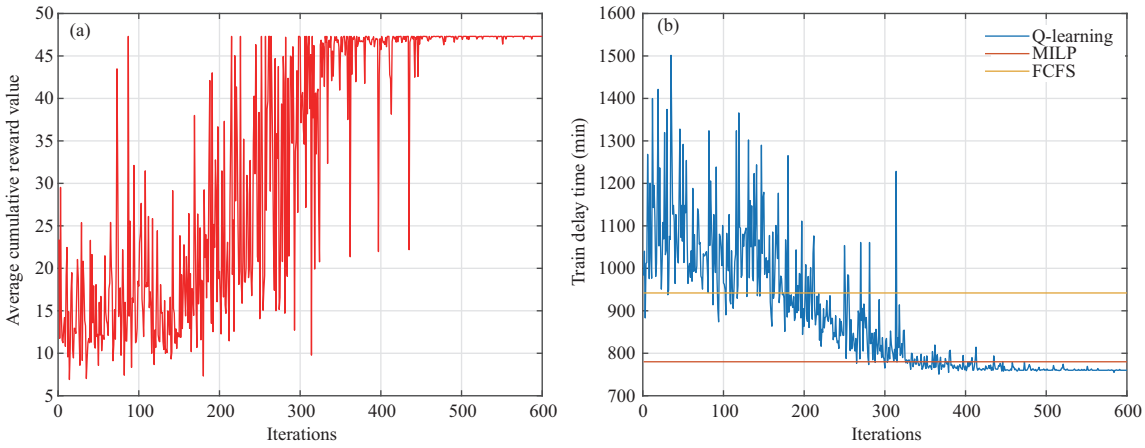

图 8　(网络版彩图) Q 学习目标函数和平均累计回报函数收敛图

Figure 8　(Color online) Convergence of the Q-learning objective function and average cumulative return function. (a) Variation of average cumulative reward value when the iteration increases; (b) variation of all trains' total delays when the iteration increases

针对上述晚点场景所提出的 Q 学习算法的训练曲线如图 8 所示, 图 8(a) 为平均累计奖励值, 大概在 400 步左右趋于稳定, 最后稳定到 7.9. 图 8(b) 曲线反映了式 (1) 目标函数 J 的收敛过程, 可以看出总晚点时间收敛至 765 min, 图中橘色、黄色曲线分别为对比算法 MILP 和 FCFS 所获得的调度方案的总晚点时间, 可以看到 Q 学习获得的调度方案具有更小的晚点时间, 尤其比 FCFS 提高明显. 在约 400 步收敛稳定后仍然有小幅波动, 这是由于求解过程中为避免陷入局部最优解, 在后期 ε 接近于 0 时仍然保持一定的随机性进行适当的探索.

Q 学习所获得的列车运行调整方案 (称作调整计划图) 如图 6 中红实线所示, 虚线为基本图, 虚线与实线重叠部分表示调整计划图与基本图一致无晚点. 该调整计划图的时空拓扑矩阵如图 7(b) 所示. 结合调整后的列车时刻表见表 2 所示, 表中加粗部分表示列车晚点, 从该表中可以看出, 对编号为 1~4, 8~10 和 13 的列车采用压缩区间运行时间进行赶点, 最后在交出站 (车站 7) 之前均恢复晚点. 列

表 2　调整后的列车时刻表

Table 2　Adjusted train timetable

Train		Station						
		ZZD	XCD	LHX	ZMDX	XYD	XGB	WH
Train1	Arrival	10:29	10:57	11:20	**11:30**	**11:50**	12:06	12:34
	Departure	10:34	10:59	11:20	**11:30**	**11:52**	12:06	12:38
Train2	Arrival	10:22	**11:16**	**11:27**	**11:35**	12:01	12:19	12:48
	Departure	10:47	**11:16**	**11:27**	**11:39**	12:03	12:19	12:51
Train3	Arrival	10:46	**11:22**	**11:33**	11:47	12:13	12:29	12:53
	Departure	10:52	**11:22**	**11:35**	11:49	12:13	12:29	13:01
Train4	Arrival	10:58	**11:38**	**11:49**	**11:59**	**12:21**	**12:34**	13:05
	Departure	11:06	**11:38**	**11:49**	**12:01**	**12:21**	**12:36**	13:08
Train5	Arrival	11:11	**11:48**	**11:59**	**12:09**	**12:44**	13:02	13:31
	Departure	11:14	**11:48**	**11:59**	**12:23**	**12:48**	13:02	13:36
Train6	Arrival	11:26	**12:00**	**12:11**	**12:21**	**12:41**	**12:54**	13:16
	Departure	11:29	**12:00**	**12:11**	**12:21**	**12:41**	**12:54**	13:19
Train7	Arrival	11:31	**12:05**	**12:16**	**12:26**	**12:46**	**12:59**	13:21
	Departure	11:36	**12:05**	**12:16**	**12:26**	**12:46**	**12:59**	13:24
Train8	Arrival	11:40	**12:17**	**12:30**	**12:56**	13:16	13:35	14:04
	Departure	11:40	**12:19**	**12:46**	**12:56**	13:18	13:35	14:13
Train9	Arrival	11:39	**12:24**	**12:52**	**13:04**	13:24	13:42	14:06
	Departure	11:45	**12:41**	**12:54**	**13:04**	13:26	13:42	14:07
Train10	Arrival	12:12	**12:47**	**13:00**	**13:12**	13:32	13:55	14:23
	Departure	12:12	**12:49**	**13:02**	**13:12**	13:39	14:00	14:27
Train11	Arrival	12:12	**12:53**	**13:06**	**13:26**	13:50	14:08	14:36
	Departure	12:18	**12:55**	**13:16**	**13:28**	13:52	14:08	14:44
Train12	Arrival	12:30	**13:03**	**13:14**	**13:24**	**13:44**	**13:57**	**14:18**
	Departure	12:33	**13:03**	**13:14**	**13:24**	**13:44**	**13:57**	**14:21**
Train13	Arrival	12:53	**13:28**	**13:39**	**13:51**	**14:11**	14:26	14:55
	Departure	12:55	**13:28**	**13:41**	**13:51**	**14:13**	14:26	14:55

车 12 由于初始晚点较大, 最后仍然晚点 1 分钟未恢复.

Q 学习的运行调整方案中, 不仅反映了采用压缩区间运行时间赶点来取得更大回报减少晚点的策略, 而且反映了合理安排越行, 调整列车运行顺序来减少晚点的策略. 如图 6(a) 中后继晚点列车 12 (三角形标注实线) 先后越行了前序列车 10 (圆圈标注实线) 和 11 (方框标注实线), 带方框虚线和实线分别为列车 11 的图定运行线和调整后的计划运行线, 从中可以看出, 列车 11 由于限速导致初始晚点 13 min, 但在车站 3 时为避让列车 12 将停站时间增加至 5 min, 使得列车 12 越行通过. 采用这种越行策略通常会导致被越行列车的晚点时间明显增加, 表现为在车站 3 的发车总晚点时间比该站的到达时间略多 (见图 9(a) 中车站 3 的到发总晚点). 虽然牺牲了被越列车 11 的短期回报 (即当前车站的晚点时间增加), 但列车 12 在越行后可通过恢复更多晚点, 全局回报 (所有列车的总晚点) 更佳. 被越列车 11, 在后继区间多次利用基本图中预留的缓冲时间赶点, 追回了由于被越行导致的晚点时间增

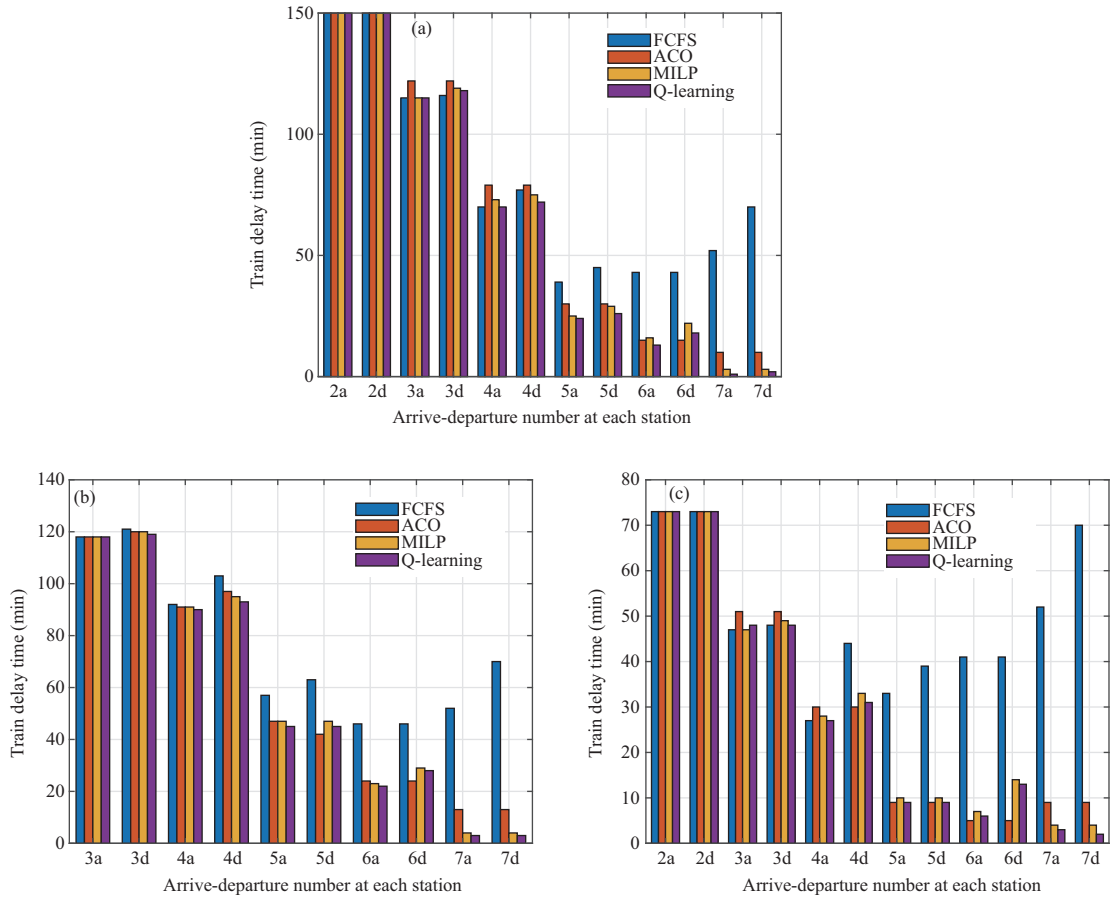

图 9 (网络版彩图) 不同典型运行场景下, 运行调整方案到发总晚点在各车站分布图

Figure 9 (Color online) Distribution of total arrival delays and total departure delays at each station under three typical railway operation scenarios. (a) Scenario 1; (b) scenario 2; (c) scenario 3

加, 在牺牲短期回报后还可以获得有利长期回报. 这种牺牲短期回报换取更佳的全局和长期回报的策略也体现在车站 6, 列车 10 被列车 12 越行. 分析表明, 本文提出的 Q 学习算法, 通过试错式交互学习, 有效地学会了利用缓冲时间和合理越行的策略, 在选择动作时不只考虑当前回报同时还考虑了长期的回报, 因此能获得一个更优的运行图.

5.2 多场景性能对比

为更全面比较本文所提 Q 学习算法的性能, 针对限速的时间以及空间范围不同, 新增两种典型场景进行验证:

场景 2 (上午 11:30 到 13:30 在车站 2 与 3 区间发生大风限速). 编号为 3~13 的列车预计到达车站 3 发生大范围晚点.

场景 3 (上午 12:00 到 13:00 在车站 1 与 2 区间发生大风限速). 编号为 8~13 的列车预计到达车站 2 发生大范围晚点.

为细化场景 1 性能比较, 将目标函数 J_a 和 J_d (16) 值分解到各个车站, 按照车站 $j = 1, \ldots, 7$ 的逐站统计到站总晚点和发车晚点的分布情况. 4 种算法得到的车站到发晚点的分布如图 9 所示, 图中

表 3　不同场景下 4 种算法的性能比较

Table 3　Performance comparison of four algorithms in different scenarios

Scenario	Number of delayed trains	Objective function value (min)			
		FCFS	ACO	MILP	Q-learning
Scenario 1	13	970	800	780	765
Scenario 2	11	768	589	578	570
Scenario 3	6	588	354	352	345

横坐标 2a 表示所有列车到达车站 2 的总晚点时间, 2d 表示从车站 2 发车的总晚点时间, 以此类推. 可以看出具备优化求解能力的 Q-learning 方法、ACO 算法和 MILP 算法求解到的运行调整计划的晚点时间随着列车运行过程总体上都是在后续车站逐渐减小, 列车运行秩序在逐渐恢复, 而 FCFS 算法在后续车站列车晚点先有减少, 但在第 5 站由于顺序不发生改变呈现晚点增加趋势. 图 9(a) 可以看出, FCFS 算法性能最差, 在第 5 站后晚点时间逐渐增加, 最终交出站晚点时间为 70 min 左右. ACO 算法最后晚点时间大于 3 min, MILP 算法最后仍有 3 min 的晚点时间未能恢复. 而本文所提算法晚点时间 (紫色) 减少的比较快, 最后在交出站第 7 站的晚点时间接近于 0.

值得指出的是, 从图 9 的 3 个子图可以看出本文所提出的 Q 学习算法在不同时空场景下所获得的运行调整方案中, 能够很好地采用不同的调度策略使得列车的晚点时间总体上逐渐减少, 并且与其他 3 个算法相比, 调度结果更优. 由此可以看出 Q 学习算法能够处理不同调度场景, 并且获得较好的调度结果.

3 种场景下, 各算法的性能指标对比见表 3, 可以看出从场景 3 到场景 1 大风限速范围从小到大, 导致的晚点列车数量和初始延误都增加, 4 种算法所获的调度方案的总晚点时间目标函数值均呈现逐步增加的趋势. 但是同一场景下 Q 学习给出的调度方案最优, 总晚点时间最少, 其次是 MILP 算法和 ACO 算法, FCFS 算法性能最差. 从上述 3 个场景中可以看出 Q 学习晚点总时间相对于 MILP 算法和 ACO 算法均减少 2% 左右, 平均每辆列车晚点时间减少 1 min 左右, 而相对于 FCFS 算法约减少 20%∼24%, 说明了本文算法的有效性.

6　结论

本文针对突发事件下高铁列车的动态调度问题, 提出了基于 Q 学习的阶段调整优化方法, 将列车的运行和调度过程建模为多阶段序列决策过程, 并定义了相应的适用于列车调度的状态空间、动作空间和回报函数. 实例分析表明, 所提 Q 学习算法能明显减少列车的总晚点时间. 本文所提无模型算法具备支持复杂、隐式约束条件的能力, 能够较好地实现联锁关系等复杂隐式约束, 在未来工作中可以在环境仿真部分集成联锁约束, 进一步提高所制定调度方案的应用可行性.

参考文献

1　Cai X, Goh C J. A fast heuristic for the train scheduling problem. Comput Oper Res, 1994, 21: 499–510

2　Cacchiani V, Huisman D, Kidd M, et al. An overview of recovery models and algorithms for real-time railway rescheduling. Transp Res Part B-Methodol, 2014, 63: 15–37

3　Zhou L S, Qin Z R. General algorithm and its realization on computer for the train operation adjustment system. J China Railway Soc, 1994, 3: 56–65 [周磊山, 秦作睿. 列车运行计划与调整的通用算法及其计算机实现. 铁道学报, 1994, 3: 56–65]

4 Dündar S, Şahin İ. Train re-scheduling with genetic algorithms and artificial neural networks for single-track railways. Transp Res Part C-Emerg Technol, 2013, 27: 1–15

5 Sato K, Tamura K, Tomii N. A MIP-based timetable rescheduling formulation and algorithm minimizing further inconvenience to passengers. J Rail Transp Planning Manage, 2013, 3: 38–53

6 Shafia M A, Aghaee M P, Sadjadi S J, et al. Robust train timetabling problem: mathematical model and branch and bound algorithm. IEEE Trans Intell Transp Syst, 2012, 13: 307–317

7 D'Ariano A, Pacciarelli D, Pranzo M. A branch and bound algorithm for scheduling trains in a railway network. Eur J Oper Res, 2007, 183: 643–657

8 Yue Y X, Wang S F, Zhou L S, et al. Optimizing train stopping patterns and schedules for high-speed passenger rail corridors. Transp Res Part C-Emerg Technol, 2016, 63: 126–146

9 Fischetti M, Monaci M. Using a general-purpose mixed-integer linear programming solver for the practical solution of real-time train rescheduling. Eur J Oper Res, 2017, 263: 258–264

10 Zhao H T. Research on optimized establishment method and evaluation method for train operation plan based on alternative graph theory. Dissertation for Ph.D. Degree. Beijing: China Academy of Railway Sciences, 2014 [赵宏涛. 基于替代图理论的列车运行计划优化编制及评价方法研究. 博士学位论文. 北京: 中国铁道科学研究院, 2014]

11 Zhao H, Dai X W. Cooperative optimization method for high-speed trains running time and energy saving based on block sections. Acta Autom Sin, 2020, 46: 77–87 [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: 77–87]

12 Xu X M, Li K P, Yang L X, et al. An efficient train scheduling algorithm on a single-track railway system. J Sched, 2019, 22: 85–105

13 Eaton J, Yang S X, Gongora M. Ant colony optimization for simulated dynamic multi-objective railway junction rescheduling. IEEE Trans Intell Transp Syst, 2017, 18: 2980–2992

14 Lin B, Yu S P, Liu Z Y, et al. High-speed railway dynamic scheduling method based on improved particle swarm algorithm. Control Eng China, 2021, 28: 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: 1334–1341]

15 Nitisiri K, Gen M, Ohwada H. A parallel multi-objective genetic algorithm with learning based mutation for railway scheduling. Comput Indust Eng, 2019, 130: 381–394

16 Feng Z Y, Cao C X, Liu Y T, et al. A multiobjective optimization for train routing at the high-speed railway station based on tabu search algorithm. Math Probl Eng, 2018, 2018: 1–22

17 Dai X W, Zhao H, Yu S P, et al. Dynamic scheduling, operation control and their integration in high-speed railways: a review of recent research. IEEE Trans Intell Transp Syst, 2021. doi: 10.1109/TITS.2021.3131202

18 Qin Z H, Li N, Liu X T, et al. Overview of research on model-free reinforcement learning. Comput Sci, 2021, 48: 180–187 [秦智慧, 李宁, 刘晓彤, 等. 无模型强化学习研究综述. 计算机科学, 2021, 48: 180–187]

19 Ning L B, Li Y D, Dong H, et al. A deep reinforcement learning approach to high-speed train timetable rescheduling under disturbances. In: Proceedings of IEEE Intelligent Transportation Systems Conference (ITSC), 2019. 3469–3474

20 Khadilkar H. A scalable reinforcement learning algorithm for scheduling railway lines. IEEE Trans Intell Transp Syst, 2019, 20: 727–736

21 Šemrov D, Marsetič R, Žura M, et al. Reinforcement learning approach for train rescheduling on a single-track railway. Transp Res Part B-Methodol, 2016, 86: 250–267

22 Zhuang H, He S W, Dai Y C. Train operation adjustment model and strategy optimization method for high speed railway. China Railway Sci, 2017, 38: 118–126 [庄河, 何世伟, 戴杨铖. 高速铁路列车运行调整的模型及策略优化方法. 中国铁道科学, 2017, 38: 118–126]

23 Liu H, Dai X W, Cui D L, et al. Optimization of high-speed train operation scheduling based on parameter adaptive improved ant colony algorithm. Control Decision, 2021, 36: 1581–1591 [刘辉, 代学武, 崔东亮, 等. 基于参数自适应蚁群算法的高速列车行车调度优化. 控制与决策, 2021, 36: 1581–1591]

Rescheduling of high-speed trains: a reinforcement learning approach

Xuewu DAI[1*], Lijuan CHENG[1], Dongliang CUI[1], Shengping YU[1], Zhiming YUAN[2] & Zhipeng YING[2]

1. *State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang* 110819*, China*;
2. *China Academy of Railway Sciences, Beijing* 100089*, China*
* Corresponding author. E-mail: daixuewu@mail.neu.edu.cn

Abstract With the construction of high-speed railway networks in China, the traffic density is constantly increasing, and the complexity and difficulty of train rescheduling sharply increase when trains are delayed due to emergencies. How to dynamically adjust the train group operation to reduce the delay and improve the punctuality rate is the core of operation adjustments. In this paper, a model-free reinforcement learning method is proposed for the operation adjustment of high-speed trains in emergency situations. Firstly, the operation adjustment of multiple trains in multiple stations and blocks is modeled as a sequential multi-stage decision-making process of constrained resource occupation and allocation, and a dynamic spatio-temporal topological matrix is proposed to model the stations and blocks. In view of the strong spatio-temporal correlation of high-speed railway trains, a reinforcement learning state space, action space and reward function containing spatio-temporal distribution information such as vehicle positions and network resources are proposed for the first time, and an effective reward feedback mechanism is constructed. Then, aiming at the difficulty of huge search space in high-speed railway operation systems, this paper proposes an adaptive generation method of heuristic action subspace. This method uses some explicit static constraints to construct heuristic rules to reduce the search space, which can effectively reduce the trial-and-error times of model-free reinforcement learning, improve the efficiency of solution, and retain the advantages of model-free generality. Finally, the simulation analysis of the case based on the actual data of Beijing-Guangzhou high-speed railway shows that the proposed algorithm can converge well and significantly reduce the delay propagation within the train group under multiple delays caused by the high wind speed limit in different space and time ranges. Compared with MILP, ACO, and FCFS algorithms, the proposed method can reduce the average delay time of the train group by 2%–20%.

Keywords reinforcement learning, spatio-temporal topology matrix, train rescheduling, FCFS algorithm, optimization