doi: 10.13413/j.cnki.jdxblxb.2025064

基于优先经验回放的生成式 SAC 算法及其应用

张 伟[1], 李玉俊[1], 谢雯雯[2], 许耘嘉[1], 孙 庚[2]
(1. 吉林大学 后勤处, 长春 130012; 2. 吉林大学 计算机科学与技术学院, 长春 130012)

摘要: 针对传统柔性演员-评论家算法在探索能力和复杂环境中状态表征不足的问题, 提出一种改进的柔性演员-评论家算法. 首先, 该算法通过引入优先经验回放机制, 利用时序差分误差对经验样本进行动态优先级评估, 从而提高关键经验的利用率, 进而提升学习效率; 其次, 该算法将生成式 Transformer 架构集成到演员网络中以增强对状态特征的动态捕捉能力, 从而显著提升其在复杂优化任务中的性能; 最后, 在高校后勤人员动态调度优化问题上进行应用实验. 实验结果表明, 与原始柔性演员-评论家算法及经典深度 Q 网络算法相比, 改进的柔性演员-评论家算法在人力需求动态拟合方面误差更小, 从而有效验证了其在实际应用中的优势和实用性.

关键词: 深度强化学习; 柔性演员-评论家算法; 优先经验回放; Transformer 架构; 后勤管理
中图分类号: TP181 **文献标志码**: A **文章编号**: 1671-5489(2025)06-1713-10

Prioritized Experience Replay-Based Generative SAC Algorithm and Its Application

ZHANG Wei[1], LI Yujun[1], XIE Wenwen[2], XU Yunjia[1], SUN Geng[2]
(1. *Logistics Department, Jilin University, Changchun* 130012, *China*;
2. *College of Computer Science and Technology, Jilin University, Changchun* 130012, *China*)

Abstract: Aiming at the problem that the conventional soft actor-critic (SAC) algorithm lacked exploration capability and state representation in complex environments, we proposed an improved soft actor-critic (ISAC) algorithm. Firstly, the ISAC algorithm introduced a prioritized experience replay (PER) mechanism, which dynamically evaluated the priority of experience samples by using the temporal differential errors, thereby enhancing the utilization of crucial experiences and improving learning efficiency of the algorithm. Secondly, the algorithm integrated generative Transformer architecture into the actor network to strengthen its ability to dynamically capture state features, thereby significantly improving its performance in complex optimization tasks. Finally, we conducted an application experiment on the dynamic scheduling optimization problem of university logistics staff. The experimental results show that, compared with the original SAC algorithm and the classic deep Q-network (DQN) algorithm, the proposed ISAC algorithm has smaller errors in dynamically fitting human resource demand, which effectively demonstrates its advantages and practicality in practical applications.

Keywords: deep reinforcement learning; soft actor-critic algorithm; prioritized experience replay;

收稿日期: 2025-02-26.

第一作者简介: 张 伟(1973—), 男, 汉族, 博士, 副研究员, 从事智能算法和后勤管理的研究, E-mail: zhangweijlu@163.com.
通信作者简介: 孙 庚(1988—), 男, 汉族, 博士, 教授, 从事生成式人工智能的研究, E-mail: sungeng@jlu.edu.cn.

基金项目: 国家自然科学基金面上项目(批准号: 62272194)和吉林大学信息化专项研究项目.

Transformer architecture; logistics management

从基于规则的算法到高级学习模型，人工智能(artificial intelligence，AI)可解决的任务变得越来越复杂[1]. 传统的AI方法，如判别式人工智能(discriminative AI，DAI)，可以从大规模数据集中学习特殊的范式，并利用深度神经网络处理分类和预测任务[2-4]. 尽管这些AI方法为现代数据驱动环境提供了基础，但其依赖大规模标注数据的特性在面对动态决策问题时存在显著局限. 在这种情况下，深度强化学习(deep reinforcement learning，DRL)通过构建智能体与环境的自主交互框架[5]，使智能体能在试错过程中直接学习最优策略，为动态系统优化提供了新方法[6-7].

在DRL领域，基于最大熵原理的柔性演员-评论家算法(soft actor-critic，SAC)因其卓越的探索能力受到广泛关注[8]. 该算法通过引入策略熵最大化的优化目标，在传统DRL的累积奖励基础上增加了策略随机性约束，从而在探索效率与策略稳定性之间实现了更好的平衡，这种特性使其在连续控制任务中展现出显著优势. 然而，SAC算法采用的经验回放机制存在固有缺陷，其均匀采样机制忽视了经验样本的价值差异性，导致具有高学习价值的样本无法获得足够的复用机会，不仅降低了样本利用效率，还可能导致策略收敛速度减缓等问题[9-10]. 此外，传统SAC的演员(actor)网络通常采用多层感知器(multi-layer perceptron，MLP)结构处理状态数据，这种结构可能难以动态捕捉状态特征之间的复杂关联[11].

针对经验回放局限性，优先级经验回放(prioritized experience replay，PER)机制为改进样本利用效率提供了一种可行方法[12-13]. 该机制建立基于时序差分误差(temporal-difference error，TD-error)的动态评估模型，通过量化样本对当前策略改进的潜在贡献度，构建非均匀采样概率分布. 此外，将PER机制融入SAC算法的经验采样过程后，还需通过重要性采样权重校正优先级，该校正机制能有效缓解因非均匀采样导致的策略估计偏差[14]. 针对状态表征方面的局限性，Transformer机制可被嵌入到输出决策的actor网络结构中，用以充分学习和建模输入的状态特征[15].

SAC等DRL算法在自动驾驶[16-18]和网络优化[19-20]等领域应用广泛. 但在后勤人员动态调度这一典型时序动态决策问题上，现有研究尚未充分挖掘其应用潜力. 该问题的复杂性主要体现在两方面：首先，人力资源需求呈现动态时变特性，因此需深入挖掘当前资源需求和员工分配之间的潜在关系；其次，解决方案必须满足多约束条件. 演化算法等传统优化算法难以在该类动态场景下提前获取先验知识，并且一旦场景发生变化，该类优化算法就需重新运行，使其不适合用于后勤人员调度优化问题的求解. 在此背景下，基于PER机制和Transformer架构的改进SAC算法(improved SAC，ISAC)更适配于后勤人员动态调度问题的求解. 基于试错学习机制的ISAC算法具备对动态环境的适应能力，且无需依赖精确的先验知识. 此外，ISAC算法通过优先级经验复用机制增强学习效率，并利用Transformer的注意力机制充分捕捉和建模环境状态特征，结合随机策略的高效探索特性，有效探索动作空间中的最优调度组合，为复杂约束下的实时调度提供了兼具自适应性与稳定性的解决方案. 本文的贡献如下：

1) 提出一种融合PER机制与Transformer架构的ISAC算法. 该算法采用PER机制重构经验采样过程，在提升训练效率的同时通过重要性采样权重更正策略更新的偏差. 此外，该算法将Transformer架构嵌入进actor网络中，借助其注意力机制增强智能体对复杂状态特征的表征能力，从而更灵活且精确地处理复杂优化问题.

2) 利用提出的ISAC算法求解高校后勤人员调度问题. 该优化问题旨在通过优化员工的调度决策，最小化调度周期内每日人力需求占比与被调度员工能力值占比之间的偏差. 实验结果表明，相较于原始SAC算法及经典深度Q网络(deep Q-network，DQN)算法，ISAC算法在动态人力需求拟合方面误差更小，显著提升了调度准确率.

1 算法设计

1.1 强化学习的基本原理

在强化学习训练过程中，智能体先在时隙t观察到当前环境状态s_t，然后执行由策略π得到的动

作 a_t，并获得相应的奖励 r_t. 强化学习的目标是使智能体获得的累计折扣奖励 G_t 最大化：

$$G_t=\sum_{\tau=t}^{T}\gamma^{\tau-1}r_\tau(s_\tau,a_\tau),\tag{1}$$

其中：$\gamma\in(0,1)$为折扣因子，表示智能体对长短期奖励的重视程度；$r_\tau(s_\tau,a_\tau)$表示智能体在状态 s_t 下执行动作 a_t 获得的奖励. 为找到能最大化累计折扣奖励的最佳策略 π^*，通常会定义一个状态动作价值函数 Q，以预估策略 π 在状态 s 下能获得的累计折扣奖励，可表示为

$$Q(s,a)=E[G_t],\tag{2}$$

其中 $E[\cdot]$表示期望.

在传统强化学习算法中，通常采用表格或简单的逼近函数估计 Q 函数. 显然，这种方式不适合处理高维状态和动作空间. 因此，提出了结合深度神经网络的 DRL 算法，以解决传统强化学习算法在高维场景下的局限性.

1.2　SAC 算法及其局限性

SAC 算法[21]因其高效的探索机制和良好的稳定性而备受关注. 不同于学习确定性策略的 DRL 算法，SAC 算法在优化目标中引入了策略熵项，以鼓励智能体在决策时保持一定程度的不确定性. 这一优化目标设计增强了智能体的探索能力. 因此，SAC 算法的优化目标可表示为

$$\max E\Big[\sum_{t=1}^{T}\gamma^{t-1}(r_t(s_t,a_t)-\rho\log\pi_\varphi(a_t|s_t))\Big],\tag{3}$$

其中：ρ 为温度系数，它控制了策略的随机程度相对于奖励的重要性；π_φ 表示参数为 φ 的 actor 网络，用于根据环境状态输出动作. 同时，SAC 算法还引入了参数分别为 θ_1 和 θ_2 的评论家(critic)网络 Q_{θ_1} 和 Q_{θ_2}，并在每次目标 Q 值计算中取两者的较小值，以抑制对动作价值的过度估计. 该机制显著提升了策略学习的稳定性和鲁棒性. 此外，$Q_{\theta'_1}$ 和 $Q_{\theta'_2}$ 分别表示两个参数为 θ'_1 和 θ'_2 的目标 critic 网络.

在 critic 网络和 actor 网络的更新阶段，传统的 SAC 算法会从经验缓冲区 B 中随机抽取一批数量为 M 的经验元组(s_t,a_t,r_t,s'_t)，用于更新神经网络的参数. critic 网络更新的损失函数可表示为

$$L(\theta_i)=\frac{1}{M}\sum_{j=0}^{M}(Q_{\theta_i}(s_j,a_j)-y_j)^2,\qquad i=1,2,\tag{4}$$

其中 y_j 表示目标值，可根据如下公式计算：

$$y_j=r_j+\gamma(\min_{i=1,2}Q_{\theta'_i}(s'_j,a'_j)-\rho\log\pi_\varphi(a'_j|s'_j)).\tag{5}$$

类似地，actor 网络更新的损失函数可表示为

$$L(\varphi)=\frac{1}{M}\sum_{j=0}^{M}(\min_{i=1,2}Q_{\theta_i}(s_j,\hat{a}_j)-\rho\log\pi_\varphi(\hat{a}_j|s_j)),\tag{6}$$

其中 $\hat{a}_j\sim\pi_\varphi(\cdot|s_j)$.

在 SAC 算法中，目标网络采用软更新方法进行参数更新. 因此，目标 critic 网络的参数更新可表示为

$$\theta'_i\leftarrow\alpha\theta'_i+(1-\alpha)\theta_i,\qquad i=1,2,\tag{7}$$

其中 α 为软更新权重，用于控制目标网络参数更新的幅度.

尽管传统 SAC 算法具有探索性强和训练稳定等诸多优势，但它仍存在如下局限性：

1) 关键经验样本利用率不足. 传统 SAC 算法使用随机经验采样机制，即算法从经验回放缓冲区随机抽取样本用于网络更新. 这种采样机制未能充分考虑每个样本对智能体学习的重要性差异，导致算法在学习过程中可能忽略某些关键经验或未充分利用那些对策略提升影响较大的经验. 因此，传统 SAC 算法在处理某些具有较高复杂度的任务时，由于无法快速聚焦于对策略改进有最大贡献的部分，导致传统 SAC 算法的学习进度通常较缓慢.

2) 状态特征建模薄弱. 传统 SAC 算法中，actor 网络使用的 MLP 结构尽管有简单和易于实现的优势，但它在处理复杂环境状态时存在明显局限. MLP 通过堆叠全连接层对输入数据进行处理，这种处理方式使 MLP 无法准确提取特征之间的时序依赖性与结构化关系. 因此，在面对 DRL 中具有复杂

关联性的环境状态时，MLP通常难以捕捉状态数据之间潜在的时序依赖性，从而影响智能体的决策精度.

1.3 ISAC算法

ISAC算法通过设计基于PER的经验采样机制和基于Transformer的策略增强机制，解决了原始SAC算法在样本利用效率和状态表征能力方面的局限性，图1为其框架示意图.

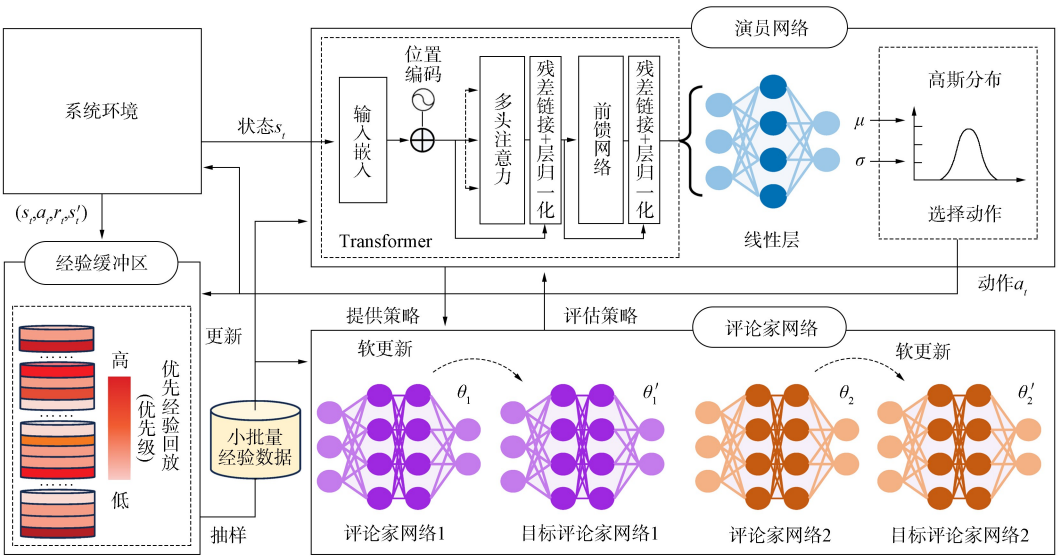

图1 ISAC算法框架示意图

Fig. 1 Schematic diagram of ISAC algorithm

1.3.1 基于PER的经验采样机制

PER机制是一种改进的经验回放策略，旨在通过赋予不同经验样本不同的优先级，从而加速DRL算法的学习过程. 因此，PER机制的核心思想是根据经验的重要性调整其被采样的概率. 其中，第k个经验被采样的概率可表示为

$$P(k)=\frac{p_k^{\alpha}}{\sum_i p_i^{\alpha}},\qquad k,i\in B, \tag{8}$$

式中$P(k)$表示第k个经验被采样的概率，α表示调节优先程度，p_k表示第k个经验的重要程度. p_k通常由TD-error衡量：

$$p_k=|\delta_k|+\dot{o}, \tag{9}$$

其中δ_k表示第k个经验的TD-error，$\dot{o}>0$是一个常数，保证p_k不为0，即保障任何一条经验都有可能被采样.

然而，TD-error大的经验样本并不总是优质样本，它们可能包含估计偏差，若不进行校正便频繁参与更新，会导致算法训练不稳定. 在这种情况下，重要性采样权重通过为每个样本赋予一个与其采样概率成反比的权重，使被频繁选中的高误差样本在算法更新中的主导作用不会被进一步放大，从而避免算法更新过程过度依赖TD-error大的经验数据，提升了训练的稳定性. 在这种情况下，第k个经验的重要性抽样权重可表示为

$$W_k=\frac{1}{S^{\beta}\cdot P(k)^{\beta}}, \tag{10}$$

其中S为经验缓冲区的大小，β为控制校正的使用程度.

因此，在采用了PER机制的SAC算法中，critic网络更新的损失函数可重新表示为

$$L(\theta_i)=\frac{1}{|B|}\sum_{j=0}^{|B|}W_j(y_j-Q_{\theta_i}(s_j,a_j))^2,\qquad i=1,2. \tag{11}$$

1.3.2　基于 Transformer 的策略增强机制

Transformer 是一种被广泛应用的生成式架构，其核心注意力机制通过全局交互模式，使序列中任意位置的元素能直接建立关联[22]. 引入 Transformer 的注意力机制后，actor 网络能显式地学习特征间的动态权重分配，从而增强对复杂状态空间的表征能力，使 ISAC 算法的策略生成过程能更深入地考虑环境状态的影响，进而优化整体决策质量.

首先，为有效建模序列数据中的位置信息，Transformer 引入了位置编码机制. 该机制使 Transformer 模型能捕捉序列中元素的相对或绝对位置关系，从而弥补了自注意力机制本身缺乏位置感知能力的不足. 位置编码机制的数学表达式如下：

$$\kappa_{2i}=\sin\left(\frac{p}{\omega^{2i/d_{\text{model}}}}\right), \tag{12}$$

$$\kappa_{2i+1}=\cos\left(\frac{p}{\omega^{2i/d_{\text{model}}}}\right), \tag{13}$$

其中 κ_{2i} 和 κ_{2i+1} 分别表示偶数和奇数位置的编码，p 表示序列中元素的绝对位置，ω 为可调节的常数参数，d_{model} 为模型隐藏层维度.

其次，自注意力机制通过动态权重分配建立序列元素间的依赖关系. 对于给定输入 $\boldsymbol{H}$，该机制先通过线性变换分别生成查询 $\boldsymbol{Q}$、键 $\boldsymbol{K}$ 和值 $\boldsymbol{V}$，其数学表达式为

$$\boldsymbol{Q}=\boldsymbol{H}\boldsymbol{W}^{\boldsymbol{Q}},\quad \boldsymbol{K}=\boldsymbol{H}\boldsymbol{W}^{\boldsymbol{K}},\quad \boldsymbol{V}=\boldsymbol{H}\boldsymbol{W}^{\boldsymbol{V}}, \tag{14}$$

其中 $\boldsymbol{H}$ 为经过位置编码后的环境状态数据，$\boldsymbol{W}^{\boldsymbol{Q}}$，$\boldsymbol{W}^{\boldsymbol{K}}$ 和 $\boldsymbol{W}^{\boldsymbol{V}}$ 表示可学习的参数矩阵. 注意力权重 $\boldsymbol{A}$ 则通过如下缩放点积注意力公式计算：

$$\boldsymbol{A}=\text{Softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{d_{\boldsymbol{K}}}}\right)\boldsymbol{V}, \tag{15}$$

其中：$d_{\boldsymbol{K}}$ 为键 $\boldsymbol{K}$ 的维度；$\dfrac{1}{\sqrt{d_{\boldsymbol{K}}}}$ 为缩放因子，用于控制点积幅值. 此外，Softmax(·)函数将权重归一化为概率分布，使模型聚焦于高相关性元素.

再次，多头注意力机制被用来进一步增强特征提取能力. 该机制分别在多个子空间独立计算注意力权重，并将各子空间的输出拼接后进行线性变换. 其完整计算过程可表示为

$$H_i=\boldsymbol{A}(\boldsymbol{H}\boldsymbol{W}_i^{\boldsymbol{Q}},\boldsymbol{H}\boldsymbol{W}_i^{\boldsymbol{K}},\boldsymbol{H}\boldsymbol{W}_i^{\boldsymbol{V}}), \tag{16}$$

$$M(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})=\text{Concat}(H_1,H_2,\cdots,H_h)\boldsymbol{W}^{\mathrm{O}}, \tag{17}$$

其中 H_i 表示第 i 个头的注意力，h 为注意力头数，Concat(·)表示将所有头的输出结果进行拼接，$\boldsymbol{W}^{\mathrm{O}}$ 为输出权重矩阵.

最后，前馈网络对注意力输出进行非线性变换和特征重整，该过程可表示为

$$F(x)=\max\{0,x\boldsymbol{W}_1+\boldsymbol{b}_1\}\boldsymbol{W}_2+\boldsymbol{b}_2, \tag{18}$$

其中 x 为从上一层获得的输出，$\boldsymbol{W}_1$ 和 $\boldsymbol{W}_2$ 表示两层线性变换的权重矩阵，$\boldsymbol{b}_1$ 和 $\boldsymbol{b}_2$ 为偏置向量.

算法 1　ISAC 算法.

输入：初始化 actor 网络、critic 网络和目标 critic 网络参数 $\varphi,\theta_1,\theta_2,\theta_1',\theta_2'$，初始化经验回放缓冲区 B，迭代次数 N_e 和时隙数量 N_T；

1) for episode=1 to N_e

2)　for t=1 to N_T

3)　　观察环境状态 $s[t]$，用基于 Transformer 的 actor 网络输出动作 $a[t]\sim\pi_\varphi(a[t]|s[t])$；

4)　　执行动作 $a[t]$，获得环境反馈奖励 $r[t]$，环境状态变为 $s'[t]$；

5)　　将经验元组$(s[t],a[t],r[t],s'[t])$存放在经验缓冲区 B 中；

6)　　根据式(8)抽取一批数量为 M 的经验数据；

7)　　根据式(10)计算重要性抽样权重；

8)　　根据式(11)更新 critic 网络 Q_{θ_1} 和 Q_{θ_2}；

9) 根据式(6)更新 actor 网络 π_φ;

10) 根据式(9)利用 TD-error 更新经验的优先级;

11) end for

12) end for

输出:actor 网络 π_φ.

1.4 ISAC 算法的计算复杂度

ISAC 算法的计算复杂度由如下四部分组成.

1) 网络初始化. 该阶段涉及 actor 网络和 critic 网络的参数初始化. 设 $|\varphi|$ 和 $|\theta|$ 分别表示 actor 网络和 critic 网络的参数数量,则网络初始化阶段的计算复杂度为 $O(|\varphi|+4|\theta|)$.

2) 动作选择. 在算法的每次迭代中,需要在每个时隙生成相应的动作. 设 N_e 和 N_T 分别表示算法的迭代总次数和每次迭代所包含的时隙数量,则动作选择阶段的计算复杂度为 $O(N_eN_T|\varphi|)$.

3) PER 机制. PER 机制包含经验存放、经验抽取及优先级更新 3 个模块. 设 C_B 和 M 分别表示经验缓冲区容量和更新批次大小,则经验抽取阶段的计算复杂度为 $O(C_B)$,优先级更新阶段的计算复杂度为 $O(M)$. 考虑到 C_B 通常远大于 M,因此 PER 机制的计算复杂度可简化为 $O(N_eN_TC_B)$.

4) 网络更新. 网络更新阶段涉及 actor 网络、双 critic 网络和目标 critic 网络的更新. 因此,网络更新阶段的计算复杂度为 $O(N_eN_T(|\varphi|+4|\theta|))$.

综上,ISAC 算法的总体计算复杂度为 $O(|\varphi|+4|\theta|+N_eN_T(|\varphi|+C_B+|\varphi|+4|\theta|))$.

2 实验结果与分析

下面通过系统性的实验设计全面评估 ISAC 算法的性能. 先在 OpenAI Gym 标准连续控制任务上进行基准测试,验证算法在连续动作空间优化问题中的性能;然后基于高校后勤人员调度问题,进一步考察 ISAC 算法在离散动作空间优化问题中的实际应用效果. 本文的实验设计涵盖了标准测试环境和实际应用场景,从而确保了对 ISAC 算法性能的多维度评估.

2.1 OpenAI Gym 基准测试

为验证 ISAC 算法的性能,选取 OpenAI Gym 环境中的 MountainCarContinuous-v0 和 LunarLanderContinuous-v2 两个典型连续控制任务进行测试.

2.1.1 环境信息

MountainCarContinuous-v0 和 LunarLanderContinuous-v2 两个环境的基本配置信息如下:

1) MountainCarContinuous-v0 的状态空间为 2 维,包含位置和速度;动作空间为 1 维,取值范围为$[-1,1]$,表示施加在小车上的推力.

2) LunarLanderContinuous-v2 的状态空间为 8 维,包括着陆器的位置、速度、角度、角速度及两个着陆脚的接触情况;动作空间为 2 维,取值范围为$[0,1]$,分别对应主引擎和侧推引擎的推力大小.

2.1.2 实验设置

在 ISAC 算法的 actor 网络中,状态数据先经过 Transformer 模块处理,然后分别通过两个独立的线性层直接输出动作分布的均值和方差. critic 网络为三层全连接的神经网络,其中包含一个具有 128 个神经元的隐藏层,激活函数为 ReLU. actor 网络和 critic 网络均采用 Adam 优化器,学习率统一设为 0.000 3,折扣因子 $\gamma=0.99$[23]. 经验缓冲区大小设为 10^6,每次更新抽取的经验批次大小为 128[24].

为验证 ISAC 算法的性能,选择如下 4 种算法作为对比算法.

1) SAC 算法:原始 SAC 算法被用来验证本文提出的两种改进因子的有效性.

2) PER-SAC 算法:PER-SAC 算法在原始 SAC 算法基础上引入 PER 机制,actor 网络仍保持 MLP 结构设计.

3) 深度确定性策略梯度(deep deterministic policy gradient, DDPG)算法:DDPG 是一种基于

actor-critic 框架的深度强化学习算法，使用确定性策略梯度和经验回放机制，适用于连续动作空间的控制问题.

4）双延迟深度确定性策略梯度(twin delayed deep deterministic policy gradient，TD3)算法：TD3 算法可视为 DDPG 算法的改进版，通过双重 critic 网络和目标策略平滑等技术减少价值估计误差，提升了算法的稳定性和性能.

2.1.3　实验结果

图 2 为不同算法在 MountainCarContinuous-v0 和 LunarLanderContinuous-v2 任务场景下获得的平均奖励值. 由图 2 可见，PER-SAC 和 ISAC 算法在两个任务中均显著优于原始 SAC 算法. PER-SAC 算法的性能提升源于所采用的 PER 机制，该机制通过重点学习高 TD-error 的经验样本，促使智能体更有效地修正策略偏差，从而获得更高的累积奖励. 此外，ISAC 算法性能相较于 PER-SAC 算法更优异，在两个任务场景下均取得了最优适应度值. 这是由于 ISAC 算法通过将 Transformer 架构整合至 actor 网络，增强了智能体对环境状态特征的提取能力，使其能更精准地捕捉状态间的关联性，从而在两个任务中均取得最优的适应度值.

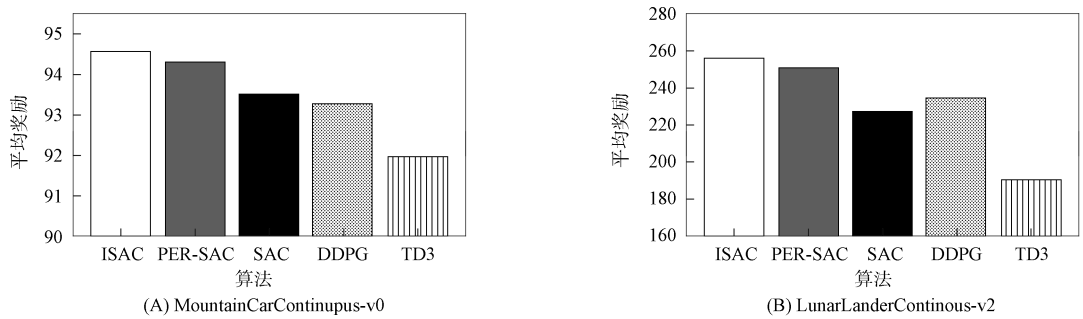

图 2　不同算法获得的平均奖励值

Fig. 2　Average reward values obtained by different algorithms

图 3 为各算法在训练过程中的奖励值收敛曲线. 由图 3 可见，各算法在训练初期性能均有较大波动，符合智能体在探索阶段通过随机动作尝试不同动作以积累经验的特点. 相较于其他对比算法，ISAC 算法和 PER-SAC 算法在训练初期展现出更快的收敛速度，验证了 PER 机制通过动态调整关键经验回放频率对学习效率的促进作用. 特别地，ISAC 算法在训练后期表现出更稳定的收敛特性，这得益于 Transformer 架构使智能体能动态适应不同环境状态，从而提升了智能体决策的质量和稳定性.

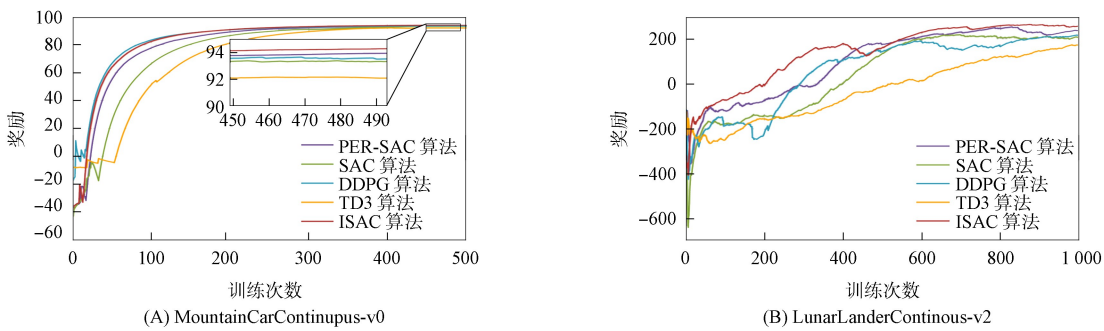

图 3　不同算法的奖励值收敛曲线

Fig. 3　Convergence curves of reward values of different algorithms

2.2　高校后勤人员调度优化

下面利用 ISAC 算法解决文献[25]中人员调度优化问题.

在场景设置方面，假设高校后勤部门每次轮休调度周期为 15 d，每位工作人员在调度周期内共休息天数为 4 d，最大连续工作天数为 4 d，最小连续工作天数为 2 d. 将后勤部门的所有员工分为 6～10 组，每组包含 3～5 名员工，组内员工同时工作或休息，每组员工在调度周期开始时的连续工作天数在最小连续和最大连续工作天数范围内随机生成. 每位员工的工作能力值由均值为 1、标准差为

0.3的分布随机生成，每日人力需求在[0.5,1.5]内随机生成.

在Markov决策过程(Markov decision process，MDP)构建方面，其核心要素定义如下：状态空间包含当日的人力需求占比、所有小组的能力值占比集合、所有小组连续工作的天数以及在人员调度过程中当日剩余的人力需求占比；动作空间表征为二元调度决策，对应每个工作组“工作/休息”状态；奖励函数设置为在每日完成最后一组工作人员的调度时的偏差指标值，即人力资源需求和实际人力资源安排之间差值的绝对值.

在算法设置方面，ISAC算法的参数设置与2.1.1节一致. 考虑到该优化问题涉及到连续工作天数约束，因此ISAC算法同样采用了文献[25]中提出的动作掩码机制以规范智能体动作，并加速学习过程. 此外，为比较ISAC算法在解决离散优化问题时的性能，选择如下两种对比算法.

1) SAC：采用原始SAC算法对高校后勤人员进行调度决策，并进行动作掩码处理.

2) DQN：鉴于DQN适合处理离散动作空间并常被用于解决调度问题，DQN算法用于对该高校后勤人员调度问题进行求解，同样对决策进行动作掩码处理.

图4为ISAC,SAC,DQN算法的收敛性能. 由图4可见，ISAC算法在收敛精度方面取得了最佳效果. 这是由于PER技术通过优先选择有更大TD误差的经验进行回放，增加了重要和罕见样本的使用频率. 在该方式下，算法能更快地关注到那些对策略改进最重要的经验，从而提高了学习效率. 此外，Transformer架构的引入增强了算法对环境状态的表征能力，从而使ISAC算法能根据环境状态做出更精准的决策. 实验结果表明，基于SAC算法的轮休调度方法比DQN算法效果更好，这是因为DQN算法的样本效率较低，通常需要更多样本才能达到同样的效果. 此外，DQN算法易出现Q值高估问题，从而导致训练波动较大，影响算法的收敛速度和效果.

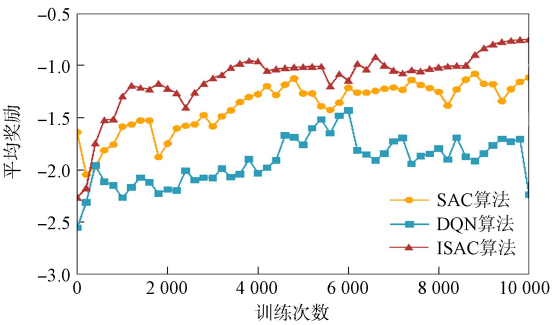

图4 不同算法在人员调度优化问题中的奖励值收敛曲线

Fig. 4 Convergence curves of reward values of different algorithms in personnel scheduling optimization problems

表1列出了不同小组数量下各算法的轮休调度偏差. 由表1可见，在平均偏差指标上，ISAC算法相较于原始SAC算法降低了45.92%，相较于DQN算法降低了63.11%. ISAC算法之所以能显著降低偏差是因为其采用了PER技术. 高TD误差样本通常表明模型在这些情况下的估计误差较大，PER通过优先处理这些样本，算法可以更快地修正这些误差，从而加速策略的优化，并提升策略的准确性. Transformer架构的引入也提高了智能体对环境状态的分析能力，从而提升了决策的准确性. 此外，实验结果表明，SAC算法取得了次优结果，这是因为SAC通过最大化奖励和策略熵之和，鼓励智能体探索多样化的行为，使SAC算法能在训练过程中持续探索，并有效避免陷入局部最优. DQN算法的效果较差主要是因为它采用了ε-贪婪策略进行探索，这种策略在探索与利用之间的平衡较简单，易陷入局部最优，尤其是当ε逐渐减小时，DQN算法更倾向于利用当前策略，减少了探索新策略的机会. 图5为ISAC算法生成的人员轮休调度表. 由图5可见，在动作掩码的作用下未产生违背约束的调度决策.

表1 不同小组数量下调度方法的偏差效果对比

Table 1 Comparison of deviation effects of scheduling methods with different numbers of groups

数量	ISAC算法	SAC算法	DQN算法	数量	ISAC算法	SAC算法	DQN算法
6组	0.006 65	0.013 28	0.020 54	9组	0.006 79	0.011 99	0.017 84
7组	0.007 25	0.012 58	0.018 47	10组	0.006 28	0.011 62	0.017 22
8组	0.006 39	0.012 28	0.016 70				

综上所述，针对传统柔性演员-评论家算法在探索能力和复杂环境中状态表征不足的问题，本文提出了ISAC算法，该算法通过PER经验采样机制解决了传统均匀采样机制产生的学习效率低问题，同

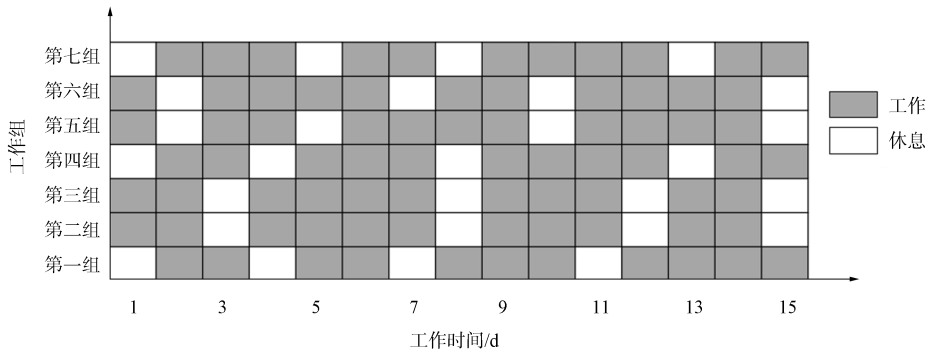

图 5　基于 ISAC 算法的人员轮休调度表

Fig. 5　ISAC algorithm-based personnel scheduling table

时通过在 actor 网络中整合 Transformer 架构增强了表征环境状态的能力. 为验证 ISAC 算法有效性，本文在 OpenAI Gym 标准连续控制任务完成基准测试后，进一步构建了高校后勤人员调度的离散动作空间应用场景进行测试. 针对人员调度的实验结果表明，在偏差指标上，本文 ISAC 算法相比基于原始 SAC 算法的轮休调度方法约降低了 45.92%，相比基于 DQN 算法的轮休调度方法约降低了 63.11%，从而验证了本文算法的有效性.

参　考　文　献

[1] ZHANG C M, LU Y. Study on Artificial Intelligence: The State of the Art and Future Prospects [J]. Journal of Industrial Information Integration, 2021, 23: 100224-1-100224-9.

[2] 傅文军，谭伟，胡露航. 从判别式人工智能到生成式人工智能的演进逻辑及场景策略研究 [J]. 中国仪器仪表，2024(10): 17-21. (FU W J, TAN W, HU L H. Study on Evolution Logic and Scenario Strategy from Discriminative Artificial Intelligence to Generative Artificial Intelligence [J]. China Instrumentation, 2024(10): 17-21.)

[3] 曹志民，张丽，郑兵，等. 基于 SMOTE 平衡数据的极端随机树岩性识别 [J]. 吉林大学学报(地球科学版)，2025, 55(4): 1372-1386. (CAO Z M, ZHANG L, ZHENG B, et al. Lithohogy Identification Using Extra Trees Based on SMOTE for Data Balancing [J]. Journal of Jilin University (Earth Science Edition), 2025, 55(4): 1372-1386.)

[4] 马世典，戴永根，江浩斌，等. 基于随机森林算法的智能转向系统故障诊断 [J]. 江苏大学学报(自然科学版)，2025, 46(5): 514-522. (MA S D, DAI Y G, JIANG H B, et al. Fault Diagnosis of Intelligent Steering System Based on Random Forest Algorithm [J]. Journal of Jiangsu University (Natural Science Edition), 2025, 46(5): 514-522.)

[5] 张一博，高丙朋. 基于深度强化学习的 AUV 路径规划研究 [J]. 东北师大学报(自然科学版)，2025, 57(1): 53-62. (ZHANG Y B, GAO B P. Research on AUV Path Planning Based on Deep Reinforcement Learning [J]. Journal of Northeast Normal University (Natural Science Edition), 2025, 57(1): 53-62.)

[6] 杨思明，单征，丁煜，等. 深度强化学习研究综述 [J]. 计算机工程，2021, 47(12): 19-29. (YANG S M, SHAN Z, DING Y, et al. Survey of Research on Deep Reinforcement Learning [J]. Computer Engineering, 2021, 47(12): 19-29.)

[7] JALALI K A Z, MANSOURI N, JAVIDI M M. Deep Reinforcement Learning-Based Scheduling in Distributed Systems: A Critical Review [J]. Knowledge and Information Systems, 2024, 66(10): 5709-5782.

[8] LIN Q S, MA H. SACHA: Soft Actor-Critic with Heuristic-Based Attention for Partially Observable Multi-agent Path Finding [J]. IEEE Robotics and Automation Letters, 2023, 8(8): 5100-5107.

[9] WEI Z G, XIAO W D, YUAN L, et al. Memory-Based Soft Actor-Critic with Prioritized Experience Replay for Autonomous Navigation [J]. Intelligent Service Robotics, 2024, 17(3): 621-630.

[10] BANERJEE C, CHEN Z, NOMAN N. Improved Soft Actor-Critic: Mixing Prioritized Off-Policy Samples with On-Policy Experiences [J]. IEEE Transactions on Neural Networks and Learning Systems, 2024, 35(3):

3121-3129.

[11] GAO Z H, WANG S X, ZHANG Z J. TSAC: Transformer-Based Soft Actor-Critic for Behavior Decision-Making in Autonomous Driving [C]//2025 4th International Conference on Artificial Intelligence, Internet and Digital Economy (ICAID). Piscataway, NJ: IEEE, 2025: 279-283.

[12] ZHANG C J, JI L H, YANG S S, et al. Distributed Optimal Consensus Control for Multiagent Systems Based on Event-Triggered and Prioritized Experience Replay Strategies [J]. Science China Information Sciences, 2025, 68(1): 1-16.

[13] FAN X K, LIU M, CHEN Y L, et al. RIS-Assisted UAV for Fresh Data Collection in 3D Urban Environments: A Deep Reinforcement Learning Approach [J]. IEEE Transactions on Vehicular Technology, 2022, 72(1): 632-647.

[14] LIU X M, ZHU T Q, JIANG C Q, et al. Prioritized Experience Replay Based on Multi-armed Bandit [J]. Expert Systems with Applications, 2022, 189: 116023-1-116023-11.

[15] WANG Z Y, GOUDARZI M, BUYYA R. TF-DDRL: A Transformer-Enhanced Distributed DRL Technique for Scheduling IoT Applications in Edge and Cloud Computing Environments [J]. IEEE Transactions on Services Computing, 2025, 18(2): 1039-1053.

[16] LIU H B, SUN J H, WANG H S, et al. Comprehensive Analysis of Adaptive Soft Actor-Critic Reinforcement Learning-Based Control Framework for Autonomous Driving in Varied Scenarios [J]. IEEE Transactions on Transportation Electrification, 2025, 11(1): 3667-3679.

[17] 刘伯鸿, 卢田. 基于 ASP-SAC 算法的列车自动驾驶速度控制 [J]. 铁道科学与工程学报, 2024, 21(7): 2637-2648. (LIU B H, LU T. Automatic Train Operation Speed Control Based on ASP-SAC Algorithm [J]. Journal of Railway Science and Engineering, 2024, 21(7): 2637-2648.)

[18] LIU H C, HUANG Z Y, MO X Y, et al. Augmenting Reinforcement Learning with Transformer-Based Scene Representation Learning for Decision-Making of Autonomous Driving [J]. IEEE Transactions on Intelligent Vehicles, 2024, 9(3): 4405-4421.

[19] HU B, MA J Q, SUN Z X, et al. Drl-Based Intelligent Resource Allocation for Physical Layer Semantic Communication with Irs [J]. Physical Communication, 2024, 63: 102270-1-102270-11.

[20] GAO X Y, SUN Y P, CHEN H, et al. Joint Computing, Pushing, and Caching Optimization for Mobile-Edge Computing Networks via Soft Actor-Critic Learning [J]. IEEE Internet of Things Journal, 2023, 11(6): 9269-9281.

[21] HAARNOJA T, ZHOU A, ABBEEL P, et al. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor [C]//Proceedings of the 35th International Conference on Machine Learning. [S.l.]: PMLR, 2018: 1861-1870.

[22] HAN K, WANG Y H, CHEN H T, et al. A Survey on Vision Transformer [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023, 45(1): 87-110.

[23] ZHAO F Y, CHEN W, LIU Z W, et al. Deep Reinforcement Learning-Based Intelligent Reflecting Surface Optimization for TDD Multi-user MIMO Systems [J]. IEEE Wireless Communications Letters, 2023, 12(11): 1951-1955.

[24] XU X L, WU F, BILAL M, et al. XRL-SHAP-Cache: An Explainable Reinforcement Learning Approach for Intelligent Edge Service Caching in Content Delivery Networks [J]. Science China Information Sciences, 2024, 67(7): 170303-1-170303-26.

[25] 李甜甜, 陈德胜, 曹斌. 基于强化学习的人员轮休调度方法 [J]. 计算机集成制造系统, 2024, 30(10): 3566-3577. (LI T T, CHEN D S, CAO B. Day-Off Scheduling Approach Based on Reinforcement Learning [J]. Computer Integrated Manufacturing Systems, 2024, 30(10): 3566-3577.)

(责任编辑: 韩 啸)