控制与决策
Control and Decision

基于多动作并行异步深度确定性策略梯度的选矿运行指标决策方法

李悄然, 丁进良

引用本文:
李悄然,丁进良. 基于多动作并行异步深度确定性策略梯度的选矿运行指标决策方法[J]. 控制与决策, 2022, 37(8): 1989–1996.

在线阅读 View online: https://doi.org/10.13195/j.kzyjc.2020.1063

基于多动作并行异步深度确定性策略梯度的选矿运行指标决策方法

李悄然, 丁进良†

(东北大学 流程工业综合自动化国家重点实验室，沈阳 110004)

摘 要: 为了解决深度确定性策略梯度算法探索能力不足的问题,提出一种多动作并行异步深度确定性策略梯度(MPADDPG)算法,并用于选矿运行指标强化学习决策. 该算法使用多个actor网络,进行不同的初始化和训练,不同程度地提升了探索能力,同时通过扩展具有确定性策略梯度结构的评论家体系,揭示了探索与利用之间的关系. 该算法使用多个DDPG代替单一DDPG,可以减轻一个DDPG性能不佳的影响,提高学习稳定性;同时通过使用并行异步结构,提高数据利用效率,加快了网络收敛速度;最后,actor通过影响critic的更新而得到更好的策略梯度. 通过选矿过程运行指标决策的实验结果验证了所提出算法的有效性.

关键词: 选矿；运行指标；决策；多动作；并行异步；深度确定性策略梯度

中图分类号: TP18 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2020.1063

引用格式: 李悄然,丁进良. 基于多动作并行异步深度确定性策略梯度的选矿运行指标决策方法[J]. 控制与决策, 2022, 37(8): 1989-1996.

Multi-action parallel asynchronous depth deterministic strategy gradient based decision-making approach of operational indices for mineral processing

LI Qiao-ran, DING Jin-liang†

(State Key Laboratory of Synthetical Automation for Process Industries，Northeastern University，Shenyang 110004，China)

Abstract: In order to solve the problem of insufficient exploration ability of the deep deterministic strategy gradient algorithm, a multi-action parallel asynchronous deep deterministic policy gradient (DDPG) algorithm is proposed for the decision-making approach of operational indices in mineral processing based on reinforcement learning. This algorithm uses multiple actor networks for different initialization and training, which greatly increases the exploration to different degrees. The relationship between exploration and utilization is revealed by extending the critical architecture of deterministic selection policy. This algorithm uses multiple DDPGs instead of a single DDPG, which can alleviate the poor performance of one DDPG and improve the learning stability. And it also improves the data utilization efficiency and speeds up the network convergence by using parallel asynchronous structure. Finally, the actor gets better strategy gradient by influencing critic's update. The effectiveness of the proposed approach has been verified by experiment results on decision-making of the operational indices in mineral processing.

Keywords: mineral processing；operational indices；decision-making；multi-actions；parallel asynchronous；deep deterministic policy gradient

0 引 言

选矿是将开采的原矿石富集为有用矿物的过程,主要目的是提高富集后精矿的品位,并实现矿物原料的高效综合利用. 选矿生产各工序的运行指标决策是协调各个工序的中间产品的质量、产量和效率等运行指标,实现选矿生产全流程精矿产量和品位的优化,保证整个生产过程目标的完成[1]. 现有的运行指标决策多采用数据驱动方法[2]、进化优化[3]、实时优化[4]或者混合方法[2]等. 文献[2]结合选矿过程运行指标优化决策,将多目标静态优化方法与综合生产指

收稿日期: 2020-07-30；**录用日期:** 2021-04-25.

基金项目: 国家重点研发计划课题(2018YFB1701104)；辽宁省科技技术项目(2020JH1/10100008).

†通讯作者. E-mail: jlding@mail.neu.edu.cn.

标预报、运行指标的前验/后验评估与动态校正相结合,提出了由运行指标初值优化、综合生产指标预报、指标前验和后验评估与动态校正组成的多目标动态智能优化决策结构和设计方法. 文献[3]针对文献[2]的运行指标初值优化问题,提出了一种多任务学习的多目标选矿运行指标进化优化算法. 文献[4]采用非线性模型预测控制和动态实时优化的DRTO双层结构,解决复杂生产过程运行优化,并在蒸馏装置上进行了验证.

然而,现有方法大多依赖数据离线建模或者精确过程模型,缺乏基于学习的智能决策方法,智能决策系统就是要实现集智能感知、控制、优化于一体,具有自适应、自学习、自动调整控制参数的功能[5]. 近年来,强化学习为运行指标智能决策提供了一种新方法,即智能体agent通过与环境不断交互,从环境中得到反馈,然后改变自身策略的方法[6]. 文献[7]提出了一种基于深度确定性策略梯度算法来处理具有连续和高维动作空间的机器人控制任务. 文献[8]提出了一种基于优先经验回放的深度确定性策略梯度算法,研究了自主水下航行器深度轨迹的深度控制问题. 文献[9]使用强化学习来求解启发式的组合优化问题,并有效地解决了在车辆路径、定向等组合优化问题中开发昂贵的问题. 这些方法为运行指标智能决策提供了有效途径.

针对选矿流程运行指标与精矿产量和品位等生产指标之间机理不清,难以用精确的机理模型描述,实际生产过程受到外部环境和内部扰动的影响,往往处于复杂多变的动态环境,运行指标决策由工艺工程师凭经验进行,人工调整不当不能保证生产指标在目标范围内等问题,本文将运行指标的决策问题转化为一个连续状态、连续动作的强化学习问题. 基于强化学习的决策方法使其具有学习能力,能够自主决策,并具有更好的自适应性与鲁棒性[10-12]. 建立具有自学习功能的智能决策系统,对实现选矿过程生产全流程优化具有重要的意义.

本文提出一种用于强化学习的连续控制的多动作并行异步深度确定性策略梯度(multi-actions parallel asynchronous deep deterministic policy gradient, MPADDPG)算法. 在连续控制中,随机策略是一个从状态到动作的概率分布的映射[13-14],相反,确定性策略是一个从状态到动作的映射[15-16]. 采用异步策略的actor-critic方法对深度确定性策略梯度算法进行改进,主要目的是解决如何在确定性策略的基础上进行决策,使agent能够更好地进行探索的问题. 利用多动作技术来改善DDPG探索能力,即创建多个参与者,每个参与者都有不同的初始化和训练,不同程度地提升了探索能力[17-18]. 同时,用于更新actor网络的策略梯度依赖于有经验的评论家,这意味着对评论家学习过程的改进可以提高actor更新的质量[19-20]. 此外,DDPG能够学习off-policy,因此可以修改收集经验的方式. 该算法并行异步地运行多个角色,所有角色都共享一个经验回放区,这使它们能够分配收集经验的任务[21-23]. 实验结果表明,所提出的MPADDPG算法改善了系统的性能.

1 问题描述

本文以选矿过程为背景,该过程采用焙烧-磨矿-磁选的工艺进行选别. 将原矿石按粒度分类进行选别,大于15 mm的矿石经焙烧还原后磁性较强,采用弱磁选机进行选别,该流程称为弱磁选别过程;小于15 mm的矿石则很难进行焙烧,细粒级的矿石磁性弱,所以采用强磁选机进行选别,该流程称为强磁选别过程. 选矿工艺过程由原矿筛分、竖炉焙烧、强/弱磁磨矿、强/弱磁选别、脱水浓缩等5个工序组成,如图1所示.

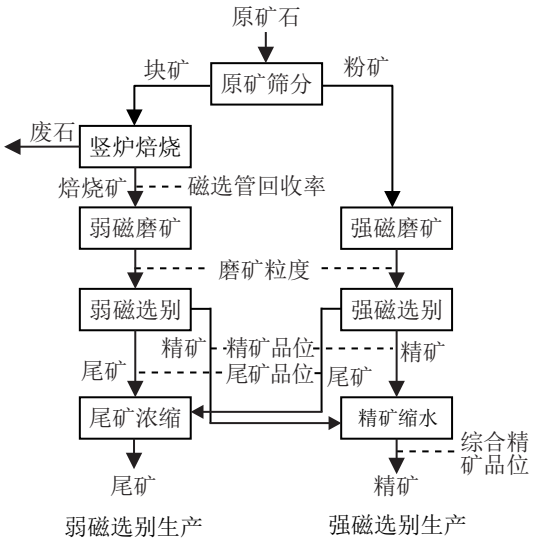


图1 选矿工艺过程

选矿过程的工况条件包括竖炉台时处理量、竖炉运时、废石品位、强磁入磨品位、弱磁入磨品位、强磁球磨机台时处理量、弱磁球磨机台时处理量、强磁球磨机运时、弱磁球磨机运时、强磁机台时处理量和弱磁机台时处理量,用$c_i(i=1,2,\ldots,M)$表示,其中M为工况数. 运行指标包括磁选管回收率、强磁粒度、弱磁粒度、强精品位、强尾品位、弱精品位和弱尾品位,用$u_{ij}(i=1,2,\ldots,I,\ j=1,2,\ldots,J)$表示,其中$I$为生产工序的个数,$J$为第$i$个生产工序的运行指标数.

运行指标决策的优化目标为综合精矿产量$\mathrm{cl}(t)$和品位$\mathrm{pw}(t)$的指标在其目标范围内,并且尽可能的高,即

$$\begin{aligned}&\max\ \mathrm{cl}(t), \max\ \mathrm{pw}(t).\\&\text{s.t. } \mathrm{cl}_{\min}\leqslant \mathrm{cl}(t)\leqslant \mathrm{cl}_{\max};\\&\qquad \mathrm{pw}_{\min}\leqslant \mathrm{pw}(t)\leqslant \mathrm{pw}_{\max}.\end{aligned}\tag{1}$$

强化学习是直接自适应最优控制[24-26],它在每个时间步长需要的计算远少于使用传统动态规划算法的间接自适应最优控制方法. 它根据马尔可夫决策过程进行建模,由智能体与选矿环境交互组成,每一时间步t,智能体在状态$\boldsymbol{s_t}=[\mathrm{cl}_t,\mathrm{pw}_t]^{\mathrm{T}}$和工况条件$\boldsymbol{c_t}=[c_{t,1},\dots,c_{t,M}]^{\mathrm{T}}$下按照策略$\pi_t$采取动作$\boldsymbol{a_t}=[u_{t,11},\dots,u_{t,1J},\dots,u_{t,IJ}]^{\mathrm{T}}$,之后根据$p(\boldsymbol{s_{t+1}}|\boldsymbol{s_t},\boldsymbol{a_t},\boldsymbol{c_t})$得到新状态$\boldsymbol{s_{t+1}}=[\mathrm{cl}_{t+1},\mathrm{pw}_{t+1}]^{\mathrm{T}}$和奖励$r_t=\mathrm{cl}_{t+1}/Q_1+\mathrm{pw}_{t+1}/Q_2$. 其中: Q_1为综合精矿产量目标值,Q_2为综合精矿品位目标值. 建立选矿过程产量模型$\mathrm{cl}_{t+1}=f_1(\mathrm{cl}_t,\boldsymbol{a_t},\boldsymbol{c_t})$和品位模型$\mathrm{pw}_{t+1}=f_2(\mathrm{pw}_t,\boldsymbol{a_t},\boldsymbol{c_t})$. 强化学习的目标是学习一种策略使累计折扣奖励的期望最大,

$$J=E\Big[\sum_{k=0}^{T}\gamma^k r_k\Big].\tag{2}$$

2　多动作并行异步DDPG算法

2.1　强化学习

强化学习基于马尔可夫决策过程(MDP)进行建模. MDP由一个智能体与环境交互组成,它包含状态空间S,动作空间A,策略$\pi(\boldsymbol{a_t}|\boldsymbol{s_t}):S\to\mathcal{P}(A)$,状态转移概率矩阵$p(\boldsymbol{s_{t+1}}|\boldsymbol{s_t},\boldsymbol{a_t})$,初始状态分布$p(\boldsymbol{s_0})$,奖励函数$r$,折扣因子$\gamma\in[0,1]$. 每一时间步$t$,智能体在状态$\boldsymbol{s_t}$按策略$\pi_t$采取动作$\boldsymbol{a_t}$,之后根据$p(\boldsymbol{s_{t+1}}|\boldsymbol{s_t},\boldsymbol{a_t})$得到新状态$\boldsymbol{s_{t+1}}$和奖励$r_t$,其中$p$是在状态$\boldsymbol{s_t}$下采取动作$\boldsymbol{a_t}$时转移到状态$\boldsymbol{s_{t+1}}$的概率. 在本文中,考虑确定性策略,即$\boldsymbol{a_t}=\pi(\boldsymbol{s_t})$.

定义折扣回报$R_t=\sum_{k=t}^{T}\gamma^{k-t}r_k$. 强化学习的目标是学习一种使期望回报最大化的策略$J^\pi=E_\pi[R_0]$. 定义动作价值函数$Q^\pi(\boldsymbol{s_t},\boldsymbol{a_t})=E_\pi[R_t|\boldsymbol{s_t},\boldsymbol{a_t}]$,式中的数学期望同时考虑了策略的随机性和环境的随机性. 定义最优策略π^*,即对于$\forall(\boldsymbol{s},\boldsymbol{a},\pi)$,满足$Q^{\pi^*}(\boldsymbol{s},\boldsymbol{a})\geqslant Q^\pi(\boldsymbol{s},\boldsymbol{a})$的任意策略$\pi^*$. 所有的最优策略都具有相同的$Q$函数,称为最优$Q$函数,记为$Q^*$. 容易证明它满足以下方程(称为Bellman方程):

$$\begin{aligned}&Q^*(\boldsymbol{s},\boldsymbol{a})=\\&E_{\boldsymbol{s'}\sim p(\cdot|\boldsymbol{s},\boldsymbol{a})}[r(\boldsymbol{s},\boldsymbol{a})+\gamma\max_{\boldsymbol{a'}}Q^*(\boldsymbol{s'},\boldsymbol{a'})|\boldsymbol{s},\boldsymbol{a}],\end{aligned}\tag{3}$$

最优策略$\pi^*=\operatorname{argmax}_{\boldsymbol{a}}Q^*(\boldsymbol{s},\boldsymbol{a})$.

强化学习算法的基本思想是估计动作价值函数,通过使用Bellman方程迭代更新$Q_{t+1}(\boldsymbol{s},\boldsymbol{a})=E_{\boldsymbol{s'}\sim p(\cdot|\boldsymbol{s},\boldsymbol{a})}[r(\boldsymbol{s},\boldsymbol{a})+\gamma\max_{\boldsymbol{a'}}Q_t(\boldsymbol{s'},\boldsymbol{a'})|\boldsymbol{s},\boldsymbol{a}]$. 这种值迭代算法收敛于最优动作价值函数$Q_t\to Q^*$当$t\to\infty$.

2.2　DDPG算法

在深度Q网络(DQN)中,通常使用神经网络函数逼近器来估计动作值函数$Q(\boldsymbol{s},\boldsymbol{a}|\theta)\approx Q^*(\boldsymbol{s},\boldsymbol{a}),\theta$为神经网络权值. 采用神经网络作为函数逼近器的DQN能够稳定、鲁棒地学习值函数. 本文算法主要有两点: 1) 利用经验缓冲区中的样本对网络进行off-policy训练,打破了样本之间的相关性,减少了更新的方差; 2) 使用同一神经网络既预测$Q(\boldsymbol{s},\boldsymbol{a}|\theta)$值又计算目标值. Q的更新容易发散,DQN为了使优化过程更加稳定,建立目标网络$Q'(\boldsymbol{s},\boldsymbol{a}|\theta')$,用于计算目标值.

DDPG是一种使用深度函数逼近器的off-policy actor-critic算法. 它基于确定性策略梯度算法[27],并结合了actor-critic方法和DQN的成功经验. Critic网络$Q(\boldsymbol{s},\boldsymbol{a}|\theta^Q)$通过最小化在每次迭代$t$时变化的损失函数$L(\theta^Q)$来训练,即

$$L(\theta^Q)=E_{\boldsymbol{s_t}\sim\rho^\beta,\boldsymbol{a_t}\sim\beta,r_t\sim E}[(Q(\boldsymbol{s_t},\boldsymbol{a_t}|\theta^Q)-y_t)^2].\tag{4}$$

其中

$$y_t=r(\boldsymbol{s_t},\boldsymbol{a_t})+\gamma Q'(\boldsymbol{s_{t+1}},\mu'(\boldsymbol{s_{t+1}}|\theta^{\mu'})|\theta^{Q'}),\tag{5}$$

ρ是与某些行为策略相关的状态访问分布,β是随机的行为策略,E是环境,y_t是目标,t表示当前时刻. 注意,目标y_t依赖于网络权值,这与监督学习使用的目标形成了对比,监督学习在学习开始前是固定的.

DDPG使用一个参数化的actor函数$\mu(\boldsymbol{s}|\theta^\mu)$,通过将状态确定地映射到特定动作来指定当前策略,并对actor网络输出的动作加入噪声$\mathcal{N}_t$,增加探索,即$\boldsymbol{a_t}=\mu(\boldsymbol{s_t}|\theta_t^\mu)+\mathcal{N}_t$. 根据actor权值对起始分布的期望回报$J$进行微分,得到如下梯度:

$$\begin{aligned}&\nabla_{\theta^\mu}J\approx E_{\boldsymbol{s_t}\sim\rho^\beta}[\nabla_{\theta^\mu}Q(\boldsymbol{s},\boldsymbol{a}|\theta^Q)|_{\boldsymbol{s}=\boldsymbol{s_t},\boldsymbol{a}=\mu(\boldsymbol{s_t}|\theta^\mu)}]=\\&E_{\boldsymbol{s_t}\sim\rho^\beta}[\nabla_{\boldsymbol{a}}Q(\boldsymbol{s},\boldsymbol{a}|\theta^Q)|_{\boldsymbol{s}=\boldsymbol{s_t},\boldsymbol{a}=\mu(\boldsymbol{s_t})}\nabla_{\theta_\mu}\mu(\boldsymbol{s}|\theta^\mu)|_{\boldsymbol{s}=\boldsymbol{s_t}}].\end{aligned}\tag{6}$$

式(6)为链式法则,通过随机梯度上升方法训练actor权值.

转换根据探索策略从环境中取样,并将转换$(\boldsymbol{s_t},\boldsymbol{a_t},r_t,\boldsymbol{s_{t+1}})$存储在经验回放区. 当经验回放区存满时,最老的样本被丢弃. 在每个时间步长中,通过从

经验回放区中均匀取样小批量数据来更新actor和critic,新的转换生成与神经网络训练交替进行. 目标网络使用软目标更新$\theta' \leftarrow \tau\theta + (1-\tau)\theta'$,其中$\tau \ll 1$. 目标网络的权值通过缓慢地跟踪学习网络来更新,防止了目标值的快速变化. 这个改变使得相对不稳定的动作价值函数学习问题更接近于监督学习的情况,大大提高了学习的稳定性.

2.3 基于DDPG的选矿过程运行指标决策

利用DDPG算法解决选矿过程运行指标决策问题,如图2所示.

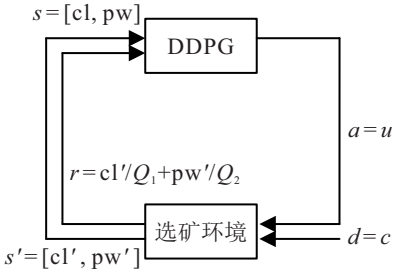


图 2 基于DDPG的选矿过程运行指标决策结构

图2中,cl为t时刻精矿产量,pw为t时刻精矿品位,$\boldsymbol{u} = [u_{11},\dots,u_{1J},\dots,u_{IJ}]^{\mathrm{T}}$为$t$时刻运行指标,$\boldsymbol{c} = [c_1,\dots,c_M]^{\mathrm{T}}$为$t$时刻工况条件,工况条件中加入随机扰动$\boldsymbol{d} = \boldsymbol{c}$. cl′为$t+1$时刻精矿产量,pw′为$t+1$时刻精矿品位.

通过马尔可夫决策过程对选矿过程进行建模,建立实际生产过程精矿产量模型$\mathrm{cl}' = f_1(\mathrm{cl}, \boldsymbol{u}, \boldsymbol{c})$和实际生产过程精矿品位模型$\mathrm{pw}' = f_2(\mathrm{pw}, \boldsymbol{u}, \boldsymbol{c})$.

2.4 多动作并行异步DDPG算法

在异步优势actor-critic(A3C)中,通常使用AC框架,包括主AC和副AC. 每个副AC都有自己的环境和学习网络副本. 这些副AC与环境交互,并以异步方式并行计算学习网络的梯度. 只有梯度是由主AC收集的,收集到的梯度用于更新学习网络,并将更新后的网络传播给每个副AC. 每个副AC都共享主AC,主AC的更新受到副AC们异步更新的不连续性影响,所以更新的相关性被降低. 本文采用并行异步的DDPG思想,与A3C不同的是,用经验回放区代替主AC,也能打消这种连续性并实现共享.

所提出的多动作并行异步DDPG算法结构如图3所示. 多动作并行异步DDPG,即MPADDPG,通过使用k个角色来并行化此过程. 该算法将经验数据的生成和选择进行了分配,k个actor生成不同的经验,并关注于actor生成的最重要的经验,将其添加到共享的经验回放区中. 每个critic从经验中抽取不同的数据来更新网络,然后更新actor网络生成不同的策略. actor与critic为单步更新,通过并行化梯度的计算来更新神经网路的参数,从而加速神经网络的训练.

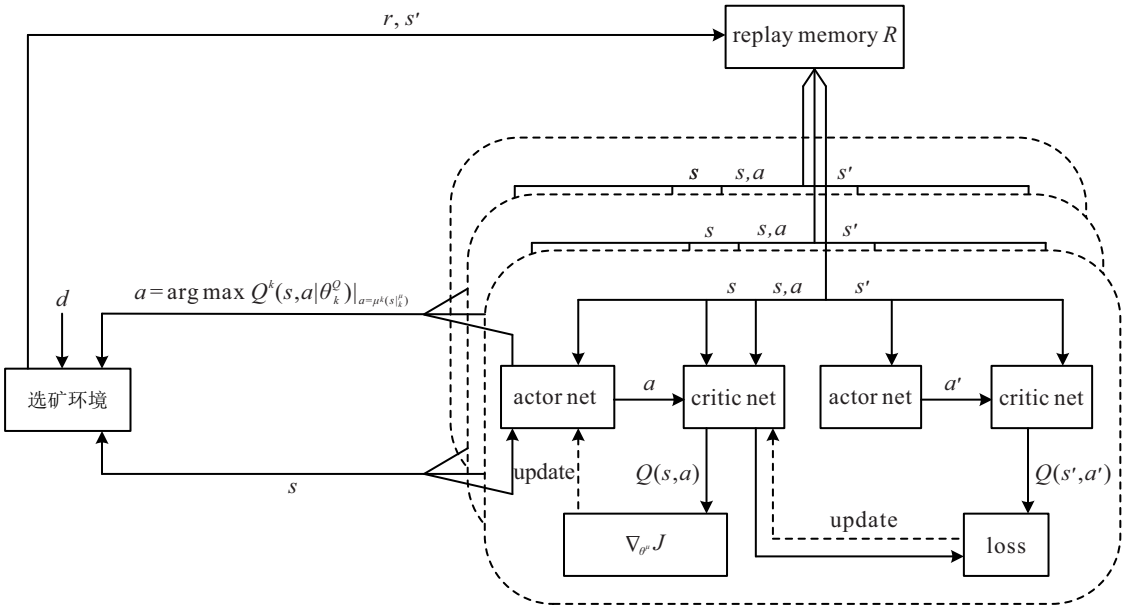


图 3 多动作并行异步DDPG算法结构

为了增加探索能力,往往在DDPG的策略网络输出动作上加入噪声,即$\boldsymbol{a_t} = \mu(\boldsymbol{s_t}|\theta_t^{\mu}) + \mathcal{N}_t$. 但由于其探索仍然不足,本文考虑了探索与利用之间的平衡,提出了一种多动作并行异步DDPG算法. 该算法扩展了DDPG的actor-critic体系结构,创建了k个actor$\{\mu_1,\dots,\mu_k\}$和critic$\{Q_1,\dots,Q_k\}$网络. 这些actor和critic被不同的θ参数化. 每个critic Q_k通过最小化误差L_k训练. 在给定状态s,动态$a = \mu_k(s)$,通过最大化$Q_k(s,a)$,采用随机梯度上升训练每个actor μ_k在每个时间步骤t,MPADDPG从所有actor

的提议中选择所对应的使Q_k值最大的动作

$$\boldsymbol{a_t} = \underset{\boldsymbol{a}}{\operatorname{argmax}} Q_k(\boldsymbol{s_t}, \boldsymbol{a}|\theta_k^Q)|_{\boldsymbol{a}=\mu_k(\boldsymbol{s_t}|\theta_k^\mu)}. \tag{7}$$

多个actor根据同一状态产生各种可能的候选动作,大大地增加了不同程度的探索. actor学习依赖于critic学习,通过扩展具有确定性策略梯度的评论家框架,使多个critic协同评估Q_k值,共同确定高潜力动作. 这种情况下,只有具有高潜力动作的经验才会被保存下来,而不是那些获得较低奖励潜力的经验,从而提高了经验回放的效率. 多DDPG的目的是为了减轻单一DDPG表现不佳的影响,以提高学习稳定性. 不同的副DDPG之间分享同一个经验回放区,进行有效的知识共享,并通过均匀抽取经验回放区中不同批量的样本进行训练. actor不是独立的,它们通过影响critic的更新而相互强化,这反过来又给了它们更好的策略梯度.

所提出的MPADDPG算法详细描述如算法1所示.

算法1　多动作并行异步DDPG算法.

随机初始化k个critic网络$Q_k(\boldsymbol{s}, \boldsymbol{a}|\theta_k^Q)$和actor网络$\mu_k(\boldsymbol{s}|\theta_k^\mu)$的参数$\theta_k^Q$和$\theta_k^\mu$;初始化目标网络$Q_k'$和$\mu_k'$的参数$\theta_k^{Q'} \leftarrow \theta_k^Q, \theta_k^{\mu'} \leftarrow \theta_k^\mu$;初始化经验缓冲区$\mathcal{R}$.

1) for episode= $1, M$;

2) 接收初始化状态$\boldsymbol{s_1}$;

3) for $t = 1, T$;

4) 选择动作$\boldsymbol{a_t} = \underset{\boldsymbol{a}}{\operatorname{argmax}} Q_k(\boldsymbol{s_t}, \boldsymbol{a}|\theta_k^Q)|_{\boldsymbol{a}=\mu_k(\boldsymbol{s_t}|\theta_k^\mu)}$;

5) 执行动作$\boldsymbol{a_t}$和$\boldsymbol{d_t}$,得到奖励r_t和下一状态$\boldsymbol{s_{t+1}}$;

6) 存储转换$(\boldsymbol{s_t}, \boldsymbol{a_t}, r_t, \boldsymbol{s_{t+1}})$于$\mathcal{R}$中;

7) 从$\mathcal{R}$中随机取样k组数量为N的转换$(\boldsymbol{s_i^k}, \boldsymbol{a_i^k}, r_i^k, \boldsymbol{s_{i+1}^k})$;

8) 设定$y_i^k = r_i^k + \gamma Q_k'(\boldsymbol{s_{i+1}^k}, \mu_k'(\boldsymbol{s_{i+1}^k}|\theta_k^{\mu'})|\theta_k^{Q'})$;

9) 更新critic网络通过最小化误差

$$L_k = \frac{1}{N}\sum_i (y_i^k - Q_k(\boldsymbol{s_i^k}, \boldsymbol{a_i^k}|\theta_k^Q))^2;$$

10) 通过如下策略梯度更新actor策略:

$$\nabla_{\theta_k^\mu} J_k \approx \frac{1}{N}\sum_i \nabla_{\boldsymbol{a^k}} Q_k(\boldsymbol{s^k}, \boldsymbol{a^k}|\theta_k^Q)|_{\boldsymbol{s^k}=\boldsymbol{s_i^k}, \boldsymbol{a^k}=\mu^k(\boldsymbol{s_i^k})} \cdot \nabla_{\theta_k^\mu}\mu_k(\boldsymbol{s^k}|\theta_k^\mu)|_{\boldsymbol{s_i^k}};$$

11) 更新目标网络

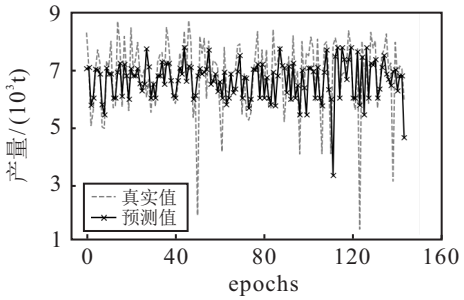
$$\theta_k^{Q'} \leftarrow \tau\theta_k^Q + (1-\tau)\theta_k^{Q'},$$
$$\theta_k^{\mu'} \leftarrow \tau\theta_k^\mu + (1-\tau)\theta_k^{\mu'}.$$

3　实验结果与分析

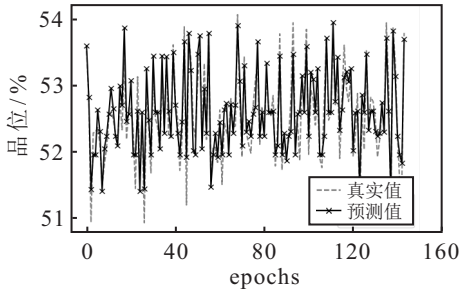
3.1　选矿运行过程建模

首先根据马尔可夫决策过程对选矿运行过程的精矿产量f_1和精矿品位f_2进行建模. 所用数据采集自中国最大的赤铁矿选矿厂生产线,共574组数据,包括生产指标$\mathrm{cl}(t)$、$\mathrm{pw}(t)$,运行指标$\boldsymbol{u}(t)$和工况条件$\boldsymbol{c}(t)$.

采用随机森林算法进行建模. 它是以决策树为基本单元,并将多棵树集成的一种算法. 算法中的每棵树随机抽样训练集,训练参数设置如下: $n\_\mathrm{estimators} = 18, \max\_\mathrm{depth} = 2, \mathrm{test\_size} = 0.25$. 所建的精矿产量$f_1$和精矿品位$f_2$模型的测试效果曲线如图4所示.



(a)　$\mathrm{cl}' = f_1(\mathrm{cl}, \boldsymbol{u}, \boldsymbol{c})$



(b)　$\mathrm{pw}' = f_2(\mathrm{pw}, \boldsymbol{u}, \boldsymbol{c})$

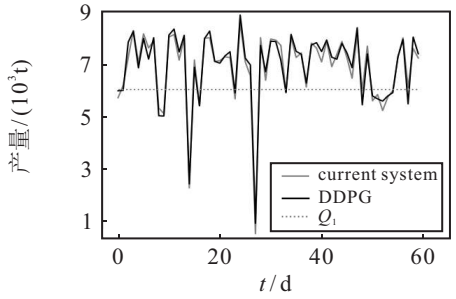
图 4　选矿过程精矿产量和品位模型测试曲线

3.2　多动作并行异步DDPG算法设置

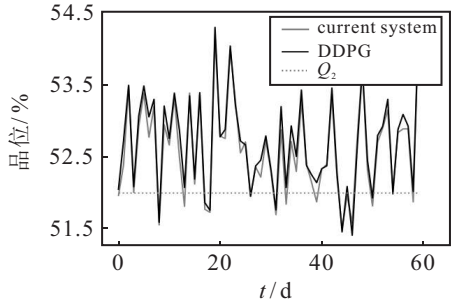
本节通过与现有系统和DDPG进行比较,验证所提出的MPADDPG算法的有效性,并对算法的可伸缩性进行分析,探讨副DDPG数量的变化对算法性能的影响. 在实验中,保持公共超参数和网络结构不变,actor和critic采用3层全连接神经网络,神经元数量分别为(200, 200, 30)个和(300, 300, 30)个,网络的权值初始化采用Xavier方法. 采用Adam优化器训练actor和critic网络,学习率分别为$1\mathrm{e}^{-5}$和$1\mathrm{e}^{-4}$,激活层都使用Relu函数. actor的输出层是tanh层,用于绑定动作至区间$[-1,1]$,折扣因子$\gamma = 0.99$,软更新$\tau = 0.001$. 使用的经验回放区大小为10^4,以$N = 64$的批量异步发送样本. 在DDPG中引入ornstein-uhlenbeck (OU)噪声,其中$\theta = 0.15, \sigma = 0.2$. 利用OU

过程产生时序相关的探索,以提高在惯性系统中的控制任务的探索效率.

如图5和表1所示,通过对现有系统与DDPG的比较结果进行分析,在60 d实验中,DDPG在提高品位的同时,显著地提高了精矿产量44.94 t. 这里将标准DDPG算法作为基线进行比较,消除了本文提出的所有增强,可以看到DDPG表现出了更好的性能. 主要原因是现有系统的运行指标是由工程师根据经验确定,操作的随机性较大,不能适应外部环境和内部干扰的动态变化. 而DDPG算法具有自学习能力,通过学习历史数据不断地调整网络参数,以选择更好的运行指标.



(a) 精矿产量



(b) 精矿品位

图5 现有系统、DDPG和目标值的精矿产量和品位比较结果

表1 现有系统与DDPG比较结果

算法	产量/t	品位/%
现有系统	6 856.74	52.59
DDPG	6 901.68	52.68

通过对DDPG与MPADDPG ($k=3$)的实验结果进行比较分析,由图6可以看出MPADDPG ($k=3$)的学习速度快于DDPG. 由图7和表2可以看出,在60 d实验中,MPADDPG ($k=3$)提高了精矿产量28.11 t,并且具有更好的性能. 原因是DDPG单角色探索不足,而MPADDPG的优势是进行多角色探索,由多个actor根据同一状态产生更多潜在的候选动作,增加了不同程度的探索,并由多个critic协同评估Q^k值,以选择Q^k值最大的动作,提高了经验回放的效率;使用多个DDPG而不是单一的DDPG,在一个DDPG表现不佳的情况下,可以削弱其影响. 同时,该算法能够在相同的时间内处理更多的数据,通过并行化用于更新参数的梯度计算,从而实现训练加速效果.

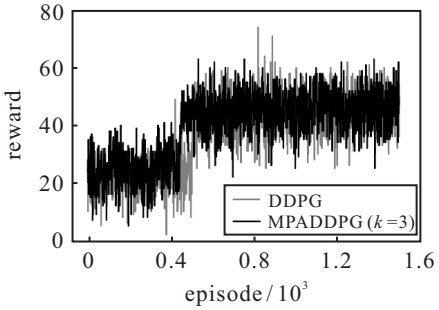


图6 DDPG与MPADDPG ($k=3$)的reward比较结果

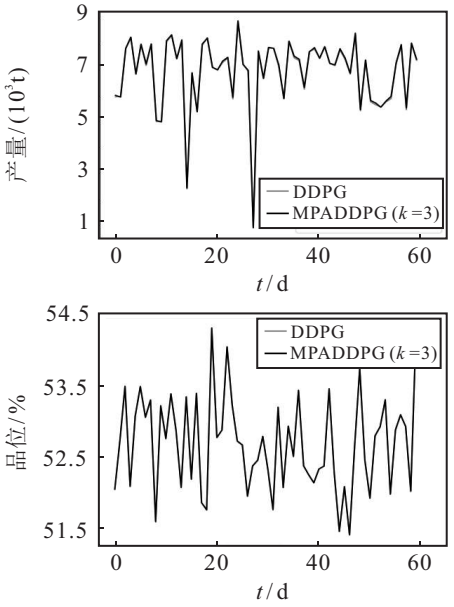


图7 DDPG与MPADDPG ($k=3$)的精矿产量和品位比较结果

表2 DDPG与MPADDPG ($k=3$)比较结果

算法	产量/t	品位/%
DDPG	6 901.68	52.68
MPADDPG ($k=3$)	6 929.80	52.68

通过对DDPG与MPADDPG ($k=5$)的实验结果进行比较分析,由图8可以看出,MPADDPG ($k=5$)学习速度快于DDPG. 由图9和表3可以看出,在60 d实验中,MPADDPG ($k=5$)提高了精矿产量35.91 t,且具有更好的性能. 通过对副DDPG数量的选择进

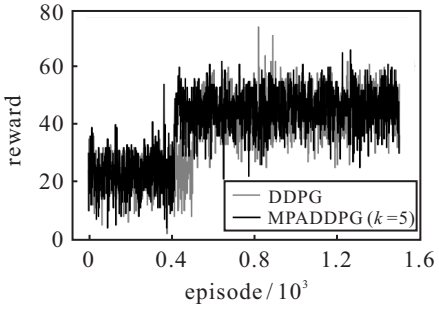


图8 DDPG与MPADDPG ($k=5$)的reward比较结果

行了讨论,发现随着副DDPG数量的增加,actor的探索能力增强,并通过影响critic的更新而相互强化,从而得到更好的策略梯度,提高了探索效率,表明了算法的性能会随着副DDPG数量的增加而不断地改进.

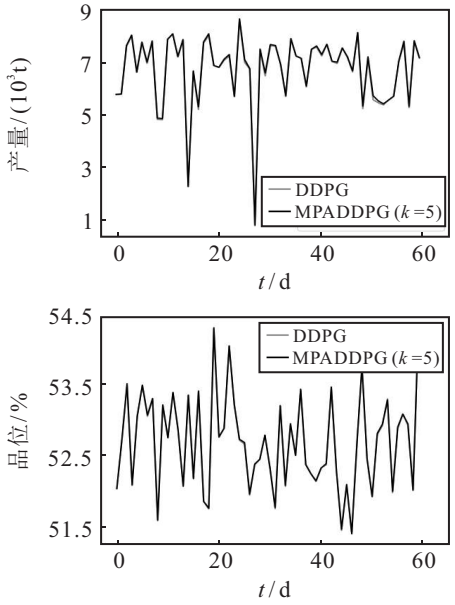


图 9　DDPG与MPADDPG ($k = 5$)的产量和品位比较结果

表 3　DDPG与MPADDPG ($k = 5$)比较结果

算法	产量/t	品位/%
DDPG	6 901.68	52.68
MPADDPG ($k = 5$)	6 937.60	52.68

为了对比不同副DDPG数量对性能的影响,分别设计了$k = 3, 5, 10$三种方式进行对比. 如表4所示,通过实验结果分析对比可知,3个版本的性能几乎相同,MPADDPG ($k = 10$)有轻微的优势. 这表明了算法的性能会随着副DDPG数量的增加而提高,但是随着副DDPG数量的盲目增加,性能的提升会非常小. 同时,考虑到并行结构是昂贵的学习系统,太多副DDPG会消耗大量的内存和时间,计算成本更高,所以应根据实际需求对副DDPG数量进行权衡,以达到最好的性能.

表 4　MPADDPG($k = 3, 5, 10$)比较结果

算法	产量/t	品位/%
MPADDPG ($k = 3$)	6 929.80	52.68
MPADDPG ($k = 5$)	6 937.60	52.68
MPADDPG ($k = 10$)	6 939.45	52.69

文献[19]运用多个评论家进行加权,得到总评分,再考虑所有行动者中评分最高者,并且保留了原始的噪声过程,较好地权衡了探索与利用问题. 将本文算法与文献[19]的方法进行了对比,结果如表5所示. 本文所提出算法的性能在$k = 5$和$k = 10$时均优于对比算法.

表 5　self-adaptive double bootstrapped DDPG与MPADDPG ($k = 3, 5, 10$)的比较结果

算法	产量/t	品位/%
MPADDPG ($k = 3$)	6 929.80	52.68
SADBDDPG	6 933.28	52.68
MPADDPG ($k = 5$)	6 937.60	52.68
MPADDPG ($k = 10$)	6 939.45	52.69

4　结　论

为了解决在选矿运行指标决策问题中当运行条件动态变化时,深度确定策略梯度算法探索能力不足的问题,本文提出了一种多动作并行异步DDPG算法. 该算法将DDPG的actor-critic架构进行了扩展,并对框架的性能进行了实验验证与对比研究. 所提出算法采用多个actor生成更多的候选动作,增加了不同程度的探索,以提高样本多样性;多个critic协同评估Q^k值,共同确定高潜力动作,以提高经验回放效率. 多DDPG代替了单一DDPG,减轻了一个DDPG表现不佳的情况,提高了学习稳定性. 本文提出的框架能够很好地适应并行角色的应用,进而有效地利用资源,加快网络训练,提高探索效率. 实验结果表明,MPADDPG算法提高了学习速度和性能,训练时间的减少与并行角色的数量大致成线性. 最后,通过选矿过程运行指标决策问题的实验结果表明了在保证精矿品位在目标范围内的情况下,提高了精矿的产量,验证了所提出算法的有效性. 下一步工作将集中在利用神经网络的回归预测值对决策结果进行校正补偿,进一步提高该方法的动态环境的适应能力.

参考文献(References)

[1] Ding J L, Chai T Y, Wang H. Offline modeling for product quality prediction of mineral processing using modeling error PDF shaping and entropy minimization[J]. IEEE Transactions on Neural Networks, 2011, 22(3): 408-419.

[2] 丁进良.动态环境下选矿生产全流程运行指标优化决策方法研究[D]. 沈阳: 东北大学, 2012.
(Ding J L. Research on optimization method of operation index of whole process of mineral processing production under dynamic environment[D]. Shenyang: Northeastern University, 2012.)

[3] Yang C E, Ding J L, Jin Y C, et al. Multitasking multiobjective evolutionary operational indices optimization of beneficiation processes[J]. IEEE Transactions on Automation Science and Engineering, 2019, 16(3): 1046-1057.

[4] Biegler L T, Yang X, Fischer G A G. Advances in sensitivity-based nonlinear model predictive control and dynamic real-time optimization[J]. Journal of Process Control, 2015, 30: 104-116.

[5] 丁进良, 杨翠娥, 陈远东, 等. 复杂工业过程智能优化决策系统的现状与展望[J]. 自动化学报, 2018, 44(11): 1931-1943.
(Ding J L, Yang C E, Chen Y D, et al. Research progress and prospects of intelligent optimization decision making in complex industrial process[J]. Acta Automatica Sinica, 2018, 44(11): 1931-1943.)

[6] Silver D, Schrittwieser J, Simonyan K, et al. Mastering the game of Go without human knowledge[J]. Nature, 2017, 550(7676): 354-359.

[7] Lillicrap T P, Hunt J J, Pritzel A, et al. Continuous control with deep reinforcement learning[EB/OL]. 2015, arXiv: 1509. 02971.

[8] Wu H, Song S J, You K Y, et al. Depth control of model-free AUVs via reinforcement learning[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2019, 49(12): 2499-2510.

[9] Kool W, van Hoof H, Welling M. Attention, learn to solve routing problems[EB/OL]. 2018, arXiv: 1803.08475.

[10] Carlini N, Wagner D. Towards evaluating the robustness of neural networks[C]. 2017 IEEE Symposium on Security and Privacy. San Jose, 2017: 39-57.

[11] Weng T W, Zhang H, Chen P Y, et al. Evaluating the robustness of neural networks: An extreme value theory approach[EB/OL]. 2018, arXiv: 1801.10578.

[12] Abdullah M A, Ren H, Ammar H B, et al. Wasserstein robust reinforcement learning[EB/OL]. 2019, arXiv: 1907.13196.

[13] Mnih V, Kavukcuoglu K, Silver D, et al. Playing atari with deep reinforcement learning[EB/OL]. 2013, arXiv: 1312.5602.

[14] Haarnoja T, Zhou A, Abbeel P, et al. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor[EB/OL]. 2018, arXiv: 1801.01290.

[15] Fujimoto S, van Hoof H, Meger D. Addressing function approximation error in actor-critic methods[EB/OL]. 2018, arXiv: 1802.09477.

[16] Wu J, Wang R, Li R Y, et al. Multi-critic DDPG method and double experience replay[C]. 2018 IEEE International Conference on Systems, Man, and Cybernetics. Miyazaki, 2018: 165-171.

[17] Plappert M, Houthooft R, Dhariwal P, et al. Parameter space noise for exploration[EB/OL]. 2017, arXiv: 1706.01905.

[18] Fortunato M, Azar G, Piot B, et al. Noisy networks for exploration[EB/OL]. 2017, arXiv: 1706.10295.

[19] Zheng Z, Yuan C, Lin Z, et al. Self-adaptive double bootstrapped DDPG[EB/OL]. [2020-07-30]. https://www.ijcai.org/proceedings/2018/0444.pdf.

[20] Zhang S T, Yao H S. ACE: An actor ensemble algorithm for continuous control with tree search[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2019, 33: 5789-5796.

[21] Barth-Maron G, Hoffman M W, Budden D, et al. Distributed distributional deterministic policy gradients[EB/OL]. 2018, arXiv: 1804.08617.

[22] Mnih V, Badia P, Mirza M, et al. Asynchronous methods for deep reinforcement learning[C]. The 33rd International Conference on Machine Learning. NewYork, 2016: 1928-1937.

[23] Horgan D, Quan J, Budden D, et al. Distributed prioritized experience replay[EB/OL]. 2018, arXiv: 1803.00933.

[24] Sutton R S, Barto A G, Williams R J. Reinforcement learning is direct adaptive optimal control[J]. IEEE Control Systems Magazine, 1992, 12(2): 19-22.

[25] Jalali A, Ferguson M. Computationally efficient adaptive control algorithms for Markov chains[C]. The 28th IEEE Conference on Decision and Control. Tampa, 1989: 1283-1288.

[26] Lewis F L, Vrabie D, Vamvoudakis K G. Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers[J]. IEEE Control Systems Magazine, 2012, 32(6): 76-105.

[27] Silver D, Lever G, Heess N, et al. Deterministic policy gradient algorithms[C].The 31st International Conference on Machine Learning. Beijing: PMLR, 2014: 387-395.

作者简介

李悄然(1996−), 女, 硕士生, 从事强化学习的研究, E-mail: qiaoranli@126.com;

丁进良(1976−), 男, 教授, 博士, 从事复杂工业过程智能建模与优化控制、生产全流程运行优化等研究, E-mail: jlding@mail.neu.edu.cn.

(责任编辑：孙艺红)