一种基于深度强化学习的海上MEC任务卸载和资源分配优化算法

徐艳丽，周子睿

（上海海事大学信息工程学院，上海 201306）

摘　要：移动边缘计算被认为是减少回程压力和提高服务质量的重要解决方案，但现有的资源管理策略在高动态的海洋环境下适应性较差。为解决该问题，提出了一种基于改进双延迟深度确定性策略梯度的任务卸载和资源分配算法。该算法可系统地协调无人机部署与边缘节点资源，联合优化通信资源分配和计算任务调度，同时考虑海洋边缘节点的能量限制和海洋网络的时变特性。具体而言，问题被表述为一个非凸优化框架，目标是在用户设备严格的服务质量要求下最大化吞吐量。提出的算法通过资源协调动态适应海洋环境变化，有效平衡了时延和能耗。仿真结果表明，在高动态的海事通信场景中，提出的算法显著优于现有的基准方法，证明该方法的有效性和可行性。

关键词：移动边缘计算；资源分配；任务卸载；海事通信；双延迟深度确定性策略梯度

中图分类号：TP393；TN92

文献标志码：A

doi: 10.11959/j.issn.1000−0801.2025227

An optimization algorithm based on deep reinforcement learning for maritime MEC task offloading and resource allocation

XU Yanli, ZHOU Zirui

College of Information Engineering, Shanghai Maritime University, Shanghai 201306, China

Abstract: Mobile edge computing is considered as an important solution to reduce backhaul pressure and improve quality of service, yet existing resource management strategies are poorly adapted in highly dynamic ocean environments. To address this problem, a task offloading and resource allocation algorithm based on an improved twin-delayed deep deterministic policy gradient was proposed. The algorithm was designed to systematically coordinate servo UAV deployment with edge node resources to jointly optimize communication resource allocation and computational task scheduling, while taking into account the energy constraints of ocean edge nodes and the time-varying

收稿日期：2025−06−30；**修回日期：**2025−09−18

通信作者：徐艳丽，ylxu@shmtu.edu.cn

基金项目：国家自然科学基金资助项目（No.62271303）；上海市教育委员会创新计划项目（No.2021-01-07-00-10-E00121）；上海市自然科学基金资助项目（No.20ZR1423200）

Foundation Items: The National Natural Science Foundation of China (No.62271303), The Innovation Program of Shanghai Municipal Education Comission of China (No.2021-01-07-00-10-E00121), The Natural Science Foundation of Shanghai (No.20ZR1423200)

characteristics of ocean networks. Specifically, the problem was formulated as a non-convex optimization framework with the objective of maximizing throughput under stringent quality of service requirements of user devices. The proposed algorithm dynamically adapted to the changing ocean environment through resource coordination, effectively balancing delay and energy consumption. Simulation results show that the proposed algorithm significantly outperforms existing benchmark methods in highly dynamic maritime communication scenarios, demonstrating the effectiveness and feasibility of the approach.

Key words: mobile edge computing, resource allocation, task offloading, maritime communication, twin-delayed deep deterministic policy gradient

0　引言

随着全球贸易的快速增长和海洋经济活动的日益复杂，海事通信技术的进步已成为支持国际航运、海洋资源勘探、海上搜救和环境监测等关键领域的必要条件[1-2]。由于浮标、监测站和船只等海洋平台的不断移动，网络拓扑结构呈现出动态和频繁变化，给海事通信带来独特的挑战。此外，海面反射、多径传播和多变的天气条件等不利影响因素会进一步降低信号质量，从而导致高干扰水平。同时，海上通信节点的稀疏分布也使在海洋环境中建立稳定、持续的覆盖变得更加复杂。

移动边缘计算（mobile edge computing，MEC）通过在最靠近用户的网络边缘处理大量数据，已成为克服这些挑战的一种颇具前景的解决方案。MEC 缩短了通信服务距离，减少了传输干扰，并显著提高了数据传输效率[3-6]。由于用户端设备可用的计算资源有限，在本地处理计算密集型任务是不切实际的。为了解决这个问题，用户服务请求通常会被卸载到附近的 MEC 节点上，从而提高响应速度并最大限度地降低时延[7]。因此，要确保快速处理用户请求，同时保持较高的服务质量（quality of service，QoS），MEC 网络中的高效资源分配至关重要。单个边缘节点的计算能力有限，且不同节点的用户请求到达率存在差异，这使得 MEC 资源的高效分配变得更加复杂。此时，需要采用协调的方法来分配通信和计算资源，以优化整体网络性能并为终端用户提供一致的 QoS[8]。

无人机具有高机动性、灵活部署能力和快速响应特性，已成为 MEC 领域的研究热点[9-11]。在地面场景中，无人机辅助的 MEC 网络可以为用户提供灵活的通信覆盖、数据采集处理和任务计算支持。因此，将无人机辅助 MEC 网络应用于海洋环境，以确保海上用户的 QoS，具有很高的应用前景。然而，海洋环境中存在节点稀疏、通信覆盖面广、环境带来的动态干扰强等一系列问题，导致实际部署面临诸多挑战[12]。首先，由于海洋节点分布稀疏且距离较远，无人机部署规划和能量管理变得尤为复杂，必须优化部署策略，同时满足覆盖和能量约束。其次，海洋环境下稀疏分布的节点通信资源匮乏，需要实现高效资源分配。最后，海洋环境中通信质量不稳定，需要考虑节点与无人机之间的可靠通信，包括通信时延和实时数据传输要求。因此，如何综合优化无人机辅助下海洋 MEC 系统的资源分配、无人机部署和计算卸载是一个亟待解决的重要问题。

虽然现有传统方案或深度强化学习（deep reinforcement learning，DRL）方案在一定程度上能够处理决策问题，但是面对海洋环境存在的高动态性仍面临适应能力不足的问题，且部分现有研究针对排队时延的建模不够精确，考虑海上高动态的通信环境，过于简化的模型无法较好地反映准确时延。同时，在无人机辅助的海洋 MEC 网络中，无人机的部署位置直接影响通信距离、

信道质量和能耗，部分研究将资源分配与无人机部署优化解耦，简单的静态部署策略无法动态响应用户分布变化，限制系统性能。基于此，本文将优化问题近似为马尔可夫决策过程（Markov decision process，MDP），提出了一种基于改进的双延迟深度确定性策略梯度（twin-delayed deep deterministic policy gradient，TD3）的联合任务卸载和资源分配（TD3-based task offloading and resource allocation，T-TORA）算法，用于优化面向海上MEC网络的资源分配、任务卸载和无人机部署策略。主要贡献有以下几点：构建了一个高动态、高干扰性的无人机辅助海上MEC网络，将TD3算法与长短期记忆（long short-term memory，LSTM）相结合，提出了一种预测模型，用于获取数据到达率和用户设备的服务率；针对在时延和能耗约束下网络吞吐量最大化的问题，提出了一种基于业务预测模型的任务卸载和资源分配算法，该算法首先基于TD3算法优化资源分配，再结合位置优化算法更新无人机位置，从而进一步提高资源分配的效率和效果。

1 相关工作

本节针对与MEC应用和部署的资源分配和计算卸载相关研究现状展开描述，主要包括传统方法和DRL方法2个方面。

1.1 基于传统方法的资源分配与计算卸载

传统方法已广泛应用于MEC应用中的优化问题。对于凸优化或可分解问题，传统优化方法可以实现全局最优，例如，文献[13]提出了一种分布式优化方案，通过联合关联用户和分配资源来减少通信和计算时延，并在仿真中实现了显著的时延和能耗性能改善。Qian等[14]采用分层算法，解决了非正交多址接入协议下边缘节点计算资源和无线资源联合分配的优化问题。Wang等[15]研究了多设备耦合通信和计算资源分配问题，引入两级串联队列模型得出串联队列的有效容量，在此基础上提出了带宽和计算资源联合分配方案，在保证QoS的同时实现网络收益最大化。针对计算时延问题，Xing等[16]提出了用户通信速率和计算频率的联合分配方案。在文献[17]中，笔者提出了一种分布式优化方案，用于多无人机协作MEC系统中的联合任务卸载决策、通信资源分配和计算资源分配，旨在最大限度地减少用户时延。Ei等[18]中提出了一种空天辅助MEC系统，该系统集成了无人机和低地轨道卫星，以实现物联网设备的高效卸载，利用块坐标下降法优化资源分配，在资源限制条件下最大限度地减少能耗和任务完成时延。在文献[19]中，笔者提出了一种针对慢速和快速衰落信道的联合任务卸载和资源分配方案，旨在任务具有顺序依赖性的物联网边缘计算场景中最大限度地降低能耗和任务时延。Wu等[20]提出了一种用于低地轨道卫星边缘计算系统任务卸载和资源分配的新策略，以最小化时延和提高资源利用率。虽然传统方法在处理此类凸和非凸优化问题时效果较好，但对于高维状态和行动空间中的多步优化问题，传统方法的效率急剧下降，甚至无法求解。

1.2 基于DRL方法的资源分配与计算卸载

DRL结合深度神经网络与强化学习，通过在复杂环境中基于状态和动作反馈不断学习得到最优策略，已被广泛应用于MEC网络中的优化问题。Guo等[21]提出了一种基于区块链的MEC框架，利用深度学习共同优化频谱分配、区块大小以及每个生产者生产的区块数量，旨在实现无线网络中的自适应资源分配。文献[22]研究了随机任务到达时的动态缓存、任务卸载和资源分配，提出了一种基于DRL的动态调度策略，以最小化长期平均成本。文献[23]介绍了一种针对MEC网络的联合部分卸载和资源分配方案，通过部署深度Q网络（deep Q network，DQN）算法优化了卸载决策和资源分配。文献[24]提出了一种基于深度确定性策略梯度（deep deterministic policy

gradient，DDPG）的联合优化算法，用于优化由可再生能源和电网能源混合供电的MEC系统的任务卸载和资源分配，其有效性得到了数值结果的证实。Yu等[25]开发了一种基于DQN的任务卸载和资源分配联合算法D-CORAL，旨在最大化MEC环境中的网络吞吐量。Zhang等[26]提出了一种基于TD3算法的优化方案，用于在支持MEC的无蜂窝大规模多输入多输出系统中进行资源分配，达到降低能耗和稳定性能的目的。在车联网领域，Hazarika等[27]提出了一种基于DRL的优先级敏感任务卸载和资源分配方案，利用软行为批判、DDPG和TD3算法在不同网络条件下实现了高效率。Liu等[28]利用双深度Q学习算法优化了卸载节点选择和资源分配，提高了MEC支持的空地一体化车载网络的效率和自主决策能力。与传统优化方法相比，DRL在处理高维、连续状态和行动空间问题时效率更高，具有更强的表达能力和泛化性能，适用于复杂环境下的决策和控制任务。

近年相关研究汇总见表1，总结了近年MEC网络资源优化问题的相关研究。与现有研究相比，本文讨论了海洋MEC网络中资源分配和任务卸载的联合优化问题。首先，与文献[17]等仅采用单一传统方案或DRL方案不同，本文采用混合方案，通过多算法协调机制在控制复杂度的前提下使系统能够同时在资源分配和无人机部署等多层面适应环境变化。其次，与文献[26]等不同，为适应节点稀疏、通信链路长程传播且网络业务高动态的海洋环境，本文针对排队时延进行了精确建模，并基于LSTM对模型关键参数进行动态预测，以优化延时估计的精度。最后，考虑问题优化的协同性，与文献[27]等不同，本文设计了一种分时尺度的协同优化机制，在高时间分辨率下的资源分配问题采用TD3进行优化，而在低时间分辨率下的无人机位置优化问题则采用鲸鱼优化算法（whale optimization algorithm，WOA）进行优化，由此，既保证了TD3学习的稳定性，又实现了无人机部署对网络动态变化的自适应调整。

表1　近年相关研究汇总

文献	解决方案	解决的问题	问题建模	优化目标	网络范式
[13]	基于定价的分布式优化	分布式任务卸载与资源分配	混合整数组合优化问题	最小化时延与电池使用惩罚的加权和	多小区MEC网络
[17]	ADMM+拉格朗日松弛	多无人机协同的MEC网络任务卸载与资源分配	非凸混合整数非线性规划	最小化总用户时延	多无人机协作的分布式MEC网络
[18]	分块坐标下降	空天一体化MEC网络任务卸载与资源分配	混合整数非凸优化问题	最小化总任务完成时延	空天一体化MEC网络
[19]	问题分解+黄金搜索法+动态规划	具有顺序任务依赖性的物联网任务卸载与资源分配	随机优化问题	最小化能耗	物联网边缘计算
[20]	DDPG	LEO卫星边缘计算中的联合卸载与频谱分配	MDP	最大化系统效用	LEO卫星边缘计算
[23]	DQN	协作式部分卸载与资源分配	MDP	最小化时延与能耗加权和	协作式物联网边缘计算
[24]	DDPG	混合能源MEC系统中的碳感知任务卸载与资源分配	MDP	最小化总系统成本	MEC赋能物联网
[25]	DQN	陆基MEC网络计算卸载与资源分配	MDP	最大化网络吞吐量	通用MEC
[26]	TD3	MEC赋能的CF-mMIMO系统中的资源分配	MDP	最小化能耗	MEC赋能CF-mMIMO
[27]	SAC+DDPG+TD3	车联网中基于优先级的任务卸载与资源分配	MDP	最大化网络平均效用	MEC赋能车联网
本文	TD3+LSTM+WOA	海上MEC网络联合任务卸载、资源分配与无人机部署	MDP+非凸优化问题	最大化网络吞吐量	无人机辅助的海上MEC网络

2 网络模型与问题建模

本文研究了1个包含多个用户设备、边缘节点、卫星、基站和数据中心的海事通信MEC网络，研究关注边缘侧优化问题，海事通信网络模型如图1所示。

2.1 网络模型

网络模型的构建假定通信船和伺服无人机均支持多种通信标准，基于模型假设通信船和伺服无人机可根据当前环境动态切换通信标准，以确保通信的稳定性和效率。首先，定义$S=\{S_n, n=1,2,\cdots,N\}$为$N$艘通信船的集合，$v=\{u_n, n=1,2,\cdots,N\}$表示$N$架伺服无人机的集合。同时，$U_e=\{U_{e,m}, m=1,2,\cdots,M\}$表示$M$个用户设备的集合。每个通信船覆盖随机分布在覆盖区域的多个用户设备。每个用户设备在一段时间内的随机时刻产生不同大小的数据请求，每个通信船为其覆盖范围内的用户设备请求提供计算服务，并为用户设备请求分配带宽资源。同时，基于软件定义网络建立虚拟中央控制器，实时观测当前通信环境中用户设备的数量、用户设备请求的大小、当前边缘设备的可用带宽资源和计算资源，同时根据观测到的信息，针对每个用户设备发出的数据请求，智能地做出最优资源分配和任务卸载决策。考虑数据中心具有强大的计算和存储能力，并且与基站相连，可以快速接收和发送数据，因此将中央控制器部署在数据中心。

假设每艘通信船拥有R_{B} MHz带宽资源和R_{C} GHz计算资源，每架伺服无人机拥有R_{U} GHz计算资源，代理根据每个设备的可用带宽资源和可用计算资源对用户设备请求进行非均匀动态资源分配。同时，假设每个用户设备请求到达通信船的频率服从泊松分布，统计到达率为$\lambda_{\mathrm{ave}}(t)$。$\alpha_k(t)$表示用户设备$k$在$t$时刻产生的请求，其特征信息可用元组$C_{\alpha_k}(t)=(D_k(t), \mathrm{Cal}_k(t), \tau_{\max})$表示，其中，$D_k(t)$表示用户设备$k$在$t$时刻产生的请求大小，$\mathrm{Cal}_k(t)$是$t$时刻的计算密度，用于衡量计算任务对计算资源和数据传输资源的相对需求，$\tau_{\max}$表示请求完成传输和计算的最大容许时延。在MEC网络中，根据各节点计算资源的可用性，采用2种不同的卸载策略来适配用户设备请求，设$S_k^n(t)$为二进制指标，定义当前用户设备请求的卸载策略：

$$S_k^n(t)=\begin{cases}0, \text{用户设备卸载至通信船}\\ 1, \text{通信船卸载至伺服无人机}\end{cases} \tag{1}$$

设$\tau_k(t)$为用户请求的总服务时延，因此总服

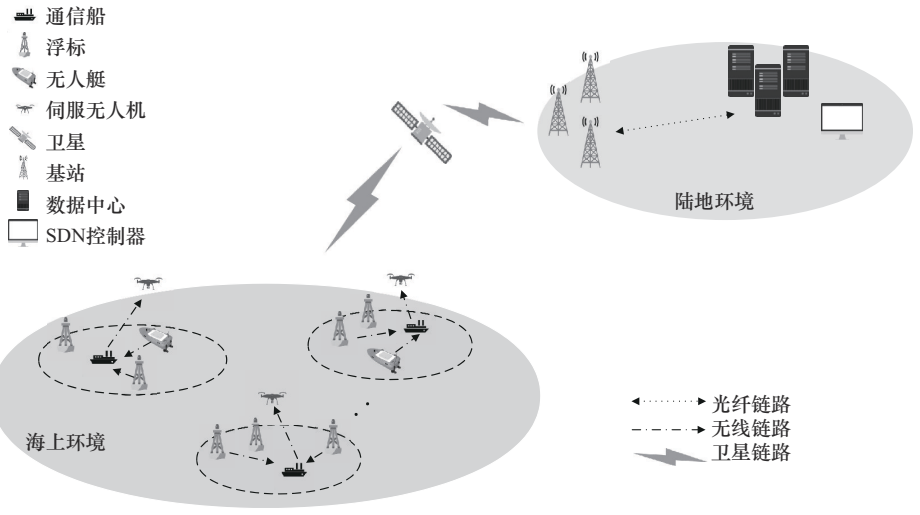


图1 海事通信网络模型

务时延的计算式为：

$$\tau_k(t)=\begin{cases}\tau_{k\_s\_1}(t), S_k^n(t)=0\\ \tau_{k\_s\_2}(t), S_k^n(t)=1\end{cases} \tag{2}$$

设置二进制指标$S_k(t)$，用于确定当前用户设备请求是否能够在最大容许时延内完成传输和计算处理，如下所示：

$$S_k(t)=\begin{cases}1, \tau_k(t)\leqslant\tau_{\max}\\ 0, \tau_k(t)>\tau_{\max}\end{cases} \tag{3}$$

2.2　接收端选择模型

假设的海事通信应用场景定义多个用户设备随机、动态地分布在通信环境中，通信设备的数量相对于用户设备数量较少，采用基于距离的K均值聚类模型来选择当前用户设备请求的接收设备，通过将距离通信设备较近的用户设备分配到同一通信设备上，降低传播时延，同时也有助于均衡通信负载的分布，避免同一设备过度拥挤，从而减少不必要的通信干扰。设$(S_{n_x}(t), S_{n_y}(t), S_{n_z}(t))$表示通信设备在时间$t$的三维坐标，即$K$-means模型中的质心，假设在环境模型的训练的一轮中的每个决策开始时基于设备坐标进行聚类以明确接入关系。同时，假设通信船有一定的通信覆盖范围，覆盖半径为R_{S_n}。初始聚类后，用户设备和候选接收设备之间的有效通信链路将根据它们之间的欧氏距离是否在此覆盖半径范围内进一步筛选，这一限制确保了只有位于可行通信范围内的用户设备才会被认为对后续的接入调度和质量评估有效。在三维环境中，用户设备在时间t时的坐标为$(\mathrm{UE}_{n_x}(t), \mathrm{UE}_{n_y}(t), \mathrm{UE}_{n_z}(t))$，作为聚类模型的输入。设$L_k(t)$为指标，若$L_k(t)=x, x\in\{1,2,\cdots,N\}$，则表示当前用户设备在通信船的覆盖范围内，并与其建立通信连接。

2.3　边缘计算模型

数据产生后经过本地网络传输到通信船并在本地网络环境中进行计算处理。首先，需要基于接收设备选择模型以确定用户数据的接收端并将数据请求传输至接收端，接收设备根据先到先服务的原则分配计算资源用于数据请求的计算处理。设$a_{\mathrm{k_comm}}(t)$表示时隙t时通信船分配给用户设备请求的带宽资源，考虑海面反射、天气突变等环境因素导致信道条件呈现时变特性，则数据请求的传输速率的计算式为：

$$r_{\mathrm{k\_trans}}(t)=a_{\mathrm{k\_comm}}(t)\mathrm{lb}\left(1+\frac{P_{\mathrm{k\_trans}}(t)H_{\mathrm{k\_env}}(t)}{P_{\mathrm{k\_env}}(t)}\right) \tag{4}$$

$$H_{\mathrm{k\_env}}(t)=H_0\cdot \mathrm{e}^{-\beta L_n(t)}\cdot\zeta(t) \tag{5}$$

其中，$P_{\mathrm{k_trans}}(t)$表示当前用户请求$\alpha_k(t)$的发射功率，设$H_{\mathrm{k_env}}(t)$为当前通信环境下用户设备与通信船之间的信道增益，可通过式（5）计算，其中，H_0为基础信道增益，由天线特性和静态环境决定；β表示路径损耗指数；$L_n(t)$表示用户设备到通信船或通信船到伺服无人机的距离；$\zeta(t)$是服从正态分布的聚合干扰，$P_{\mathrm{k_env}}(t)=N_{\mathrm{env}}a_{\mathrm{k_comm}}(t)$是当前通信环境中的噪声功率，其中，$N_{\mathrm{env}}$为环境噪声功率谱密度。由此，用户设备数据请求$\alpha_k(t)$的传输时延可计算如下：

$$\tau_{\mathrm{k\_trans}}(t)=\frac{D_k(t)R_{\mathrm{o}}^k(t)}{r_{\mathrm{k\_trans}}(t)}+\tau_{\mathrm{pro\_delay\_n}}(t) \tag{6}$$

其中，$R_{\mathrm{o}}^k(t)$为用户请求$\alpha_k(t)$的卸载率，定义为在时隙t时，用户数据请求分配给执行节点的比例，取值区间为[0，1]，当$R_{\mathrm{o}}^k(t)=0$时，表示该数据请求不卸载，仅在本地处理，当$R_{\mathrm{o}}^k(t)=1$时，则表示该数据请求全部卸载至执行节点；$\tau_{\mathrm{pro_delay_n}}(t)$为用户设备与通信船之间的传播时延，设$L_{\mathrm{u_c_n}}(t)$为用户设备与通信船间的距离，同时假设无线传播速度为光速c，则传播时延为：

$$\tau_{\mathrm{pro\_delay\_n}}(t)=\frac{L_{\mathrm{u\_c\_n}}(t)}{c} \tag{7}$$

当$R_{\mathrm{B_n}}^{\mathrm{ava}}(t)<a_{\mathrm{k_comm}}(t)$时，用户设备请求需要进入等待队列等待资源释放，设$W_1$表示平均排队等待时延。应用排队理论中的通用/通用/1（general/general/1，G/G/1）模型估计排队等待时

延，基于G/G/1模型的排队等待时延的近似计算可以应用Kingman公式[29]，由此可得平均排队等待时延W_1的计算式为：

$$W_1 \approx \frac{\rho}{1-\rho} \cdot \frac{c_a^2+c_s^2}{2} \cdot \frac{1}{\mu} \tag{8}$$

其中，系统利用率由$\rho=\lambda/\mu$给出，λ和μ分别是平均到达率和平均服务率，c_a^2和c_s^2分别是到达间隔平方变异系数和服务时间平方变异系数。设$a_{\mathrm{k_comp_CS}}(t)$为分配给用户设备请求$\alpha_k(t)$的计算资源，而当$R_{\mathrm{C_CS_n}}^{\mathrm{ava}}(t) \geqslant a_{\mathrm{k_comp_CS}}(t)$时，当前用户设备请求的计算时延为：

$$\tau_{\mathrm{k\_comp}}(t)=\frac{D_k(t)R_{\mathrm{o}}^k(t)\mathrm{Cal}_k(t)}{a_{\mathrm{k\_comp\_CS}}(t)} \tag{9}$$

因此，当$S_k^n(t)=0$时，当前用户设备请求完成传输和计算的总时延为：

$$\tau_{\mathrm{k\_s\_1}}(t)=\begin{cases}\tau_{\mathrm{k\_trans}}(t)+\tau_{\mathrm{k\_comp}}(t), R_{\mathrm{B\_n}}^{\mathrm{ava}}(t) \geqslant a_{\mathrm{k\_comm}}(t)\\ \tau_{\mathrm{k\_trans}}(t)+\tau_{\mathrm{k\_comp}}(t)+W_1, R_{\mathrm{B\_n}}^{\mathrm{ava}}(t) < a_{\mathrm{k\_comm}}(t)\end{cases} \tag{10}$$

当$R_{\mathrm{C_CS_n}}^{\mathrm{ava}}(t) < a_{\mathrm{k_comp_CS}}(t)$时，通信船会将当前用户设备请求卸载给伺服无人机处理。在这种情况下，传输时延中也需要考虑传播时延，计算如下：

$$\tau'_{\mathrm{pro\_delay\_n}}(t)=\frac{L_{\mathrm{c\_s\_n}}(t)}{c} \tag{11}$$

其中，$L_{\mathrm{c_s_n}}(t)$为通信船与伺服无人机之间的距离。因此，数据传输总时延的计算式如下：

$$\tau'_{\mathrm{k\_trans}}(t)=\frac{D_k(t)R_{\mathrm{o}}^k(t)}{r_{\mathrm{k\_trans}}(t)}+\tau'_{\mathrm{pro\_delay\_n}}(t) \tag{12}$$

当$R_{\mathrm{C_SUAV_n}}^{\mathrm{ava}}(t) < a_{\mathrm{k_comp_SUAV}}(t)$时，用户设备请求$\alpha_k(t)$需要进入等待队列等待资源释放，设$W_2$为排队等待时延。应用排队理论中的G/G/1模型估算排队等待时延。因此，用户设备请求在伺服无人机上完成计算卸载和计算的总时延可按式（13）求得：

$$\tau_{\mathrm{k\_s\_2}}(t)=\begin{cases}\tau'_{\mathrm{k\_trans}}(t)+\tau_{\mathrm{k\_comp}}(t), R_{\mathrm{C\_SUAV\_n}}^{\mathrm{ava}}(t) \geqslant a_{\mathrm{k\_comp\_SUAV}}(t)\\ \tau'_{\mathrm{k\_trans}}(t)+\tau_{\mathrm{k\_comp}}(t)+W_2, R_{\mathrm{C\_SUAV\_n}}^{\mathrm{ava}}(t) < a_{\mathrm{k\_comp\_SUAV}}(t)\end{cases} \tag{13}$$

2.4 伺服无人机能耗模型

当$S_k^n(t)=1$时，用户设备请求被卸载到伺服无人机上，并利用它来补偿计算资源，此时需要考虑伺服无人机的能耗。基于假设的海事通信环境和通信过程，悬停能耗$E_{\mathrm{k_n_hover}}(t)$和飞行能耗$E_{\mathrm{k_n_move}}(t)$是伺服无人机能耗的主要部分。因此，总能耗的计算式为：

$$E_{\mathrm{k\_n}}(t)=E_{\mathrm{k\_n\_hover}}(t)+E_{\mathrm{k\_n\_move}}(t) \tag{14}$$

考虑伺服无人机在向用户设备请求提供计算资源进行计算处理时处于悬停状态，由此可通过式（15）、式（16）计算无人机的悬停能耗：

$$P_{\mathrm{suav\_n\_hover}}(t)=\sqrt{\frac{(m\cdot g)^3}{2A\rho_{\mathrm{env}}(t)}} \tag{15}$$

$$E_{\mathrm{k\_n\_hover}}(t)=P_{\mathrm{suav\_n\_hover}}(t)\cdot T_{\mathrm{comp}}(t) \tag{16}$$

其中，$P_{\mathrm{suav_n_hover}}(t)$表示伺服无人机在悬停时的功耗，$m$为伺服无人机的质量，$g$为重力加速度，$\rho_{\mathrm{env}}(t)$为环境中的空气密度，$A$是伺服无人机旋翼的总扫掠面积，$T_{\mathrm{comp}}(t)$是伺服无人机计算过程中消耗的总时延，存在以下2种不同的取值情况：

$$T_{\mathrm{comp}}(t)=\begin{cases}\tau_{\mathrm{k\_comp}}(t), R_{\mathrm{C\_SUAV\_n}}^{\mathrm{ava}}(t) \geqslant a_{\mathrm{k\_comp\_SUAV}}(t)\\ \tau_{\mathrm{k\_comp}}(t)+W_2, R_{\mathrm{C\_SUAV\_n}}^{\mathrm{ava}}(t) < a_{\mathrm{k\_comp\_SUAV}}(t)\end{cases} \tag{17}$$

接下来，考虑无人机移动到求解出的最优位置所消耗的能耗，计算式如下：

$$P_{\mathrm{suav\_n\_move}}(t)=P_{\mathrm{suav\_n\_hover}}(t)\cdot(1+k\cdot v_{\mathrm{suav\_n}}^3) \tag{18}$$

$$E_{\mathrm{k\_n\_move}}=P_{\mathrm{suav\_n\_move}}(t)\cdot T_{\mathrm{flight\_n}}(t) \tag{19}$$

其中，$P_{\mathrm{suav_n_move}}(t)$表示伺服无人机飞行功耗，$k$为无人机飞行功耗调整系数，$v_{\mathrm{suav_n}}$为无人机飞行速度，$T_{\mathrm{flight_n}}(t)=L'(t)/v_{\mathrm{suav_n}}$为飞行时间，其

中，$L'(t)$是伺服无人机初始位置与最佳位置之间的距离。

2.5　问题建模

本文旨在通过优化通信和计算资源的分配，并结合LSTM和WOA来辅助资源分配，在总时延和伺服无人机能耗的约束条件下，最大化系统中用户设备请求的总吞吐量。由此，相应的优化问题表述如下：

$$\max_{S_k^n(t),a_{\text{k\_comm}}(t),a_{\text{k\_comp\_CS}}(t),a_{\text{k\_comp\_SUAV}}(t),R_o^k(t)} \sum_{\alpha_k(t)} S_k(t)$$

限制条件如下所示：

$$\text{C}_7: R_B/4 \leqslant a_{\text{k\_comm}}(t) \leqslant R_B/2, \forall k \quad (20a)$$

$$\text{C}_8: R_C/4 \leqslant a_{\text{k\_comp\_CS}}(t) \leqslant R_C/2, \forall k \quad (20b)$$

$$\text{C}_9: R_U/4 \leqslant a_{\text{k\_comp\_SUAV}}(t) \leqslant R_U/2, \forall k \quad (20c)$$

$$\text{C}_{10}: 0 \leqslant R_{\text{o}}^k(t) \leqslant 1, \forall k \quad (20d)$$

$$\text{C}_{11}: \sum_{n=1}^{N} S_k^n(t) = 1, S_k^n(t) \in \{0,1\}, \forall k,n \quad (20e)$$

$$\text{C}_{12}: \tau_{\text{coh}} \geqslant \tau_{\text{k\_trans}}(t) + \tau_{\text{pro\_delay}}(t) \quad (20f)$$

$$\text{C}_{13}: \tau_k(t) \leqslant \tau_{\max} \quad (20g)$$

$$\text{C}_{14}: E_k(t) \leqslant E_{\max} \quad (20h)$$

其中，约束条件C_7和C_8分别是通信船对用户设备请求的带宽资源和计算资源分配的约束，既确保资源最小效用，又可避免过度占用，以实现负载均衡和公平性。C_9是伺服无人机计算资源分配的约束条件。C_{10}为用户设备请求卸载率的限制。C_{11}是对卸载次数的限制，即每个用户请求最多只能卸载1次。C_{12}是信道特性保持稳定的最大时间间隔，确保在信道一致性间隔内完成传输，在实际部署时，需要结合海上信道特性来动态调整τ_{coh}，具体来说，SDN控制器根据节点位置差分估算相对速度，利用最大多普勒频移关系实时计算τ_{coh}，并在雨衰气象数据超过阈值时引入衰减因子进一步缩短其取值。C_{13}为时延约束，C_{14}是对伺服无人机能耗的限制，其中，$E_{\max}$代表伺服无人机可容忍的最大能耗。

3　T-TORA算法

在本节中，基于构建的海事MEC网络模型，将在时延和能量消耗的约束下实现网络吞吐量最大化这一非凸优化问题近似为MDP。随后，提出了一种基于改进TD3算法的方案来解决这一问题。T-TORA算法结构如图2所示，分为3个部分：环境观测层、学习决策层和优化层。

3.1　环境观测层

代理在时隙t时接收到来自动态环境的观测值$O_{i,t}$并生成观测空间$O(t)$，由此可得：

$$O(t)=\{U_I(t),P_S(t),P_U(t),P_{\text{Sa}}(t),P_G(t),R_B^{\text{ava}}(t),R_{\text{C\_CS}}^{\text{ava}}(t),R_{\text{C\_SUAV}}^{\text{ava}}(t),I_{\text{env}}(t)\} \quad (21)$$

其中，$U_I(t)=\{(I_k(t),D_k(t),\mathbf{P}_k(t))\}$表示用户设备请求$\alpha_k(t)$的相关信息，$I_k(t)$为用户ID，$D_k(t)$为时隙$t$时产生的请求数据大小，$P_k(t)$为时隙$t$时用户设备的位置坐标；$P_{\text{S}}(t)$为时隙$t$时通信船位置坐标集合；$P_{\text{U}}(t)$为时隙$t$时无人机位置坐标集合；$P_{\text{Sa}}(t)$为时隙$t$时卫星位置坐标；$P_{\text{G}}(t)$为时隙$t$时地面基站位置坐标；$R_{\text{B}}^{\text{ava}}(t)=\{R_{\text{B_1}}^{\text{ava}}(t),\cdots,R_{\text{B_}n}^{\text{ava}}(t)\}$表示通信船可用带宽集合；$R_{\text{C_CS}}^{\text{ava}}(t)=\{R_{\text{C_CS_1}}^{\text{ava}}(t),\cdots,R_{\text{C_CS_n}}^{\text{ava}}(t)\}$表示通信船可用计算资源集合；$R_{\text{C_SUAV}}^{\text{ava}}(t)=\{R_{\text{C_SUAV_1}}^{\text{ava}}(t),\cdots,R_{\text{C_SUAV_n}}^{\text{ava}}(t)\}$表示伺服无人机可用计算资源集合；$I_{\text{env}}(t)$为环境噪声。下面，代理接收到的环境观测数据被转换成输入向量，并输入学习决策层。

3.2　学习决策层

首先，该层的Actor网络接收输入的初始环境观测值s_t，经过多个全连接层，并使用激活函数ReLU，因此每一层的输出计算如下所示：

$$\boldsymbol{h}_i^{\pi}=f(\boldsymbol{W}_i^{\pi}\boldsymbol{h}_{i-1}^{\pi}+\boldsymbol{b}_i^{\pi}) \quad (22)$$

其中，$\boldsymbol{W}_i^{\pi}$和$\boldsymbol{b}_i^{\pi}$分别为第i层的权重矩阵和偏置向量；f为ReLU激活函数；$\boldsymbol{h}_{i-1}^{\pi}$为第$i-1$层的输出，同时第1层的输入为状态向量$\boldsymbol{s}_t$，即$\boldsymbol{h}_0^{\pi}=\boldsymbol{s}_t$。最后一层的输出经过线性变换后得到动作向量

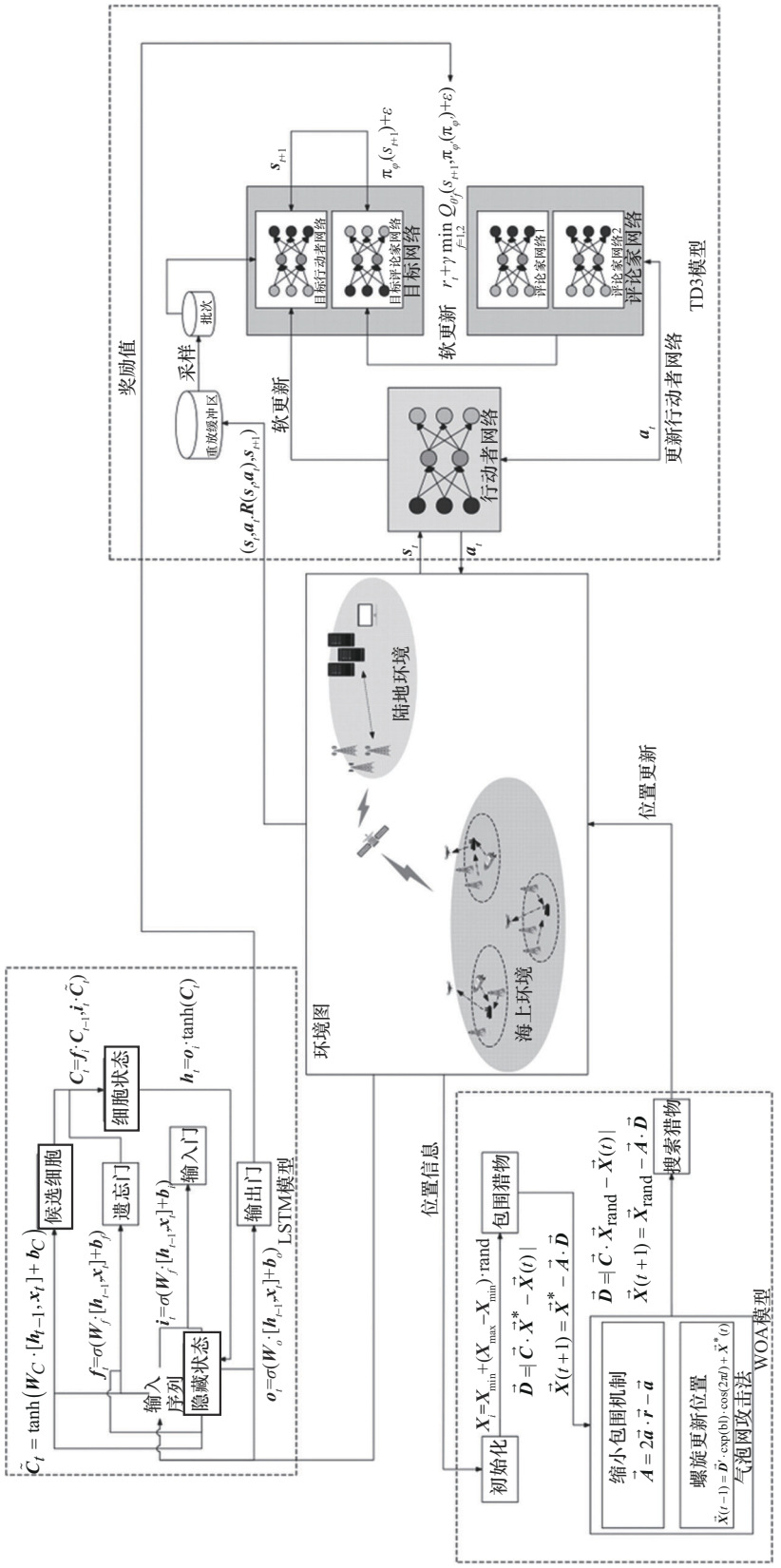


图2 T-TORA算法结构

$a_t = \{S_k^n(t), a_{\text{k\_comm}}(t), a_{\text{k\_comp\_CS}}(t), a_{\text{k\_comp\_SUAV}}(t), R_o^k(t)\}$，由此可得：

$$\boldsymbol{a}_t = \pi_{\boldsymbol{\varphi}}(\boldsymbol{s}_t) = \tanh(\boldsymbol{W}_n^{\pi}\boldsymbol{h}_{n-1}^{\pi} + \boldsymbol{b}_n^{\pi}) \tag{23}$$

其中，$\boldsymbol{W}_n^{\pi}$ 和 $\boldsymbol{b}_n^{\pi}$ 分别为第 n 层的权重矩阵和偏置向量；$\boldsymbol{h}_{n-1}^{\pi}$ 为倒数第二层的输出。接下来根据动作向量计算奖励值 $R(s_t, a_t)$，同时更新下一个状态向量 $\boldsymbol{s}_{t+1}$，由此奖励值为：

$$R(s_t, a_t) = \begin{cases} d_k^t(t) + d_k^e(t) + d_k^a(t), \tau_k(t) \leqslant \tau_{\max} \\ R_{\text{minus}}, \tau_{\text{coh}} \geqslant \tau_k(t) > \tau_{\max} \\ R'_{\text{minus}}, \tau_k(t) > \tau_{\text{coh}} \end{cases} \tag{24}$$

其中，$d_k^t(t) = \dfrac{\tau_{\max} - \tau_k(t)}{\tau_{\max}}$；$d_k^e(t) = \dfrac{E_{\max} - E_k(t)}{E_{\max}}$；$d_k^a(t) = -0.5\left(\dfrac{2a_{\text{k\_comm}}(t)}{R_B} + \dfrac{2a_{\text{k\_comp\_CS}}(t)}{R_C} + \dfrac{2a_{\text{k\_comp\_SUAV}}(t)}{R_U}\right)$；$R_{\text{minus}}$ 为负数，记为−10，R'_{minus} 为−20。下面，将 $\boldsymbol{s}_t$，$\boldsymbol{a}_t$，$R(\boldsymbol{s}_t, \boldsymbol{a}_t)$ 和 $\boldsymbol{s}_{t+1}$ 合并为 1 个元组 $(\boldsymbol{s}_t, \boldsymbol{a}_t, R(\boldsymbol{s}_t, \boldsymbol{a}_t), \boldsymbol{s}_{t+1})$，存入经验重放缓冲区，并从经验重放缓冲区中随机抽取一批数据用于更新网络参数。同时，该层的 Critic 网络根据输入的状态向量 $\boldsymbol{s}_t$ 和 $\boldsymbol{a}_t$ 计算 Q 值，以此来评估 Actor 网络生成的动作。具体而言，在接收到输入的状态向量和动作向量后，Critic 网络将状态向量和动作向量进行拼接得到输入向量 $\boldsymbol{X}_t = [\boldsymbol{s}_t, \boldsymbol{a}_t]$，经过几个全连接层，同时配合激活函数 ReLU，得到每一层的输出。第 1 层的输入向量为 $\boldsymbol{X}_t = [\boldsymbol{s}_t, \boldsymbol{a}_t]$，即 $\boldsymbol{h}_0^{\pi} = \boldsymbol{X}_t$。最后一层的输出经过线性变换得到 Q 值，如下所示：

$$Q(\boldsymbol{s}_t, \boldsymbol{a}_t) = \boldsymbol{W}_n^{Q}\boldsymbol{h}_{n-1}^{Q} + \boldsymbol{b}_n^{Q} \tag{25}$$

其中，$\boldsymbol{W}_n^{Q}$ 和 $\boldsymbol{b}_n^{Q}$ 分别代表第 n 层的权重矩阵和偏置向量，$\boldsymbol{h}_{n-1}^{Q}$ 为倒数第二层的输出。此外，TD3 算法使用双 Q 值网络来缓解 Q 值过高估计的问题，从而抑制单个网络中由于噪声或过拟合导致的估计偏差。具体而言，目标 Actor 网络生成下一个动作向量 $\boldsymbol{a}_{t+1} = \pi_{\boldsymbol{\varphi}'}(\boldsymbol{s}_{t+1}) + \varepsilon$，然后使用 2 个目标 Critic 网络计算 Q 值，如下所示：

$$y_i = r_i + \gamma \min_{j=1,2} Q_{\boldsymbol{\theta}_j'}(\boldsymbol{s}_{t+1}, \pi_{\boldsymbol{\varphi}'}(\boldsymbol{s}_{t+1}) + \varepsilon) \tag{26}$$

其中，γ 为折扣因子；$\boldsymbol{\theta}_j'$ 为目标 Critic 网络 j 的参数；$\boldsymbol{\varphi}'$ 为目标 Actor 网络的参数；ε 为策略噪声，用于增加策略的稳定性。最后，利用 2 个 Critic 网络计算损失值，并通过梯度下降法更新 Critic 网络参数，具体如下所示：

$$L(\boldsymbol{\theta}_j) = E[(Q_{\boldsymbol{\theta}_j'}(\boldsymbol{s}_i, \boldsymbol{a}_i) - y_i)^2] \tag{27a}$$

$$\nabla_{\boldsymbol{\theta}_j} L(\boldsymbol{\theta}_j) = \nabla_{\boldsymbol{\theta}_j} E[(Q_{\boldsymbol{\theta}_j'}(\boldsymbol{s}_i, \boldsymbol{a}_i) - y_i)^2] \tag{27b}$$

$$\boldsymbol{\theta}_j \leftarrow \boldsymbol{\theta}_j - \alpha \nabla_{\boldsymbol{\theta}_j} L(\boldsymbol{\theta}_j) \tag{27c}$$

其中，α 为学习率。下面基于此计算该层 Actor 网络的损失，并利用梯度下降法更新 Actor 网络的参数，通过最小化损失值来最大化 Critic 网络的 Q 值，具体如下：

$$L(\boldsymbol{\varphi}) = -E[Q_{\boldsymbol{\theta}_1}(\boldsymbol{s}_t, \pi_{\boldsymbol{\varphi}}(\boldsymbol{s}_{t+1}))] \tag{28a}$$

$$\nabla_{\boldsymbol{\varphi}} L(\boldsymbol{\varphi}) = -E[\nabla_{\boldsymbol{\varphi}} Q_{\boldsymbol{\theta}_1}(\boldsymbol{s}_t, \pi_{\boldsymbol{\varphi}}(\boldsymbol{s}_{t+1}))] \tag{28b}$$

$$\boldsymbol{\varphi} \leftarrow \boldsymbol{\varphi} - \alpha \nabla_{\boldsymbol{\varphi}} L(\boldsymbol{\varphi}) \tag{28c}$$

根据上述步骤，代理在每个时间步获取环境观测数据，并通过对经验重放缓冲区中的批量数据进行采样，更新 Actor 网络和 Critic 网络的参数。Actor 网络负责生成行动，Critic 网络负责评估行动的价值。策略噪声和探索噪声指导网络的训练，实现对环境的学习，以及资源分配和计算卸载的决策。最后，为了稳定训练过程，TD3 算法采用软更新方法来更新目标网络的参数，如下所示：

$$\boldsymbol{\varphi}' \leftarrow \tau\boldsymbol{\varphi} + (1-\tau)\boldsymbol{\varphi}' \tag{29}$$

$$\boldsymbol{\theta}_j' \leftarrow \tau\boldsymbol{\theta}_j + (1-\tau)\boldsymbol{\theta}_j' \tag{30}$$

其中，τ 为软更新率；$\boldsymbol{\varphi}$ 和 $\boldsymbol{\theta}_j$ 分别为当前 Actor 网络和 Critic 网络的参数；$\boldsymbol{\varphi}'$ 和 $\boldsymbol{\theta}_j'$ 分别为目标 Actor 网络和 Critic 网络的参数。此外，考虑 LSTM 的时序依赖和小样本学习动态建模能力，本文在学习决策层引入了 LSTM 模块。具体而言，在每个时间步，SDN 控制器会记录每艘通信船接收到的用户请求数量和处理完成的请求数量，以及每架无人机接收到的卸载数据请求和处理完成的请求数量，基于此，计算得出若干个时间步的平均任

务到达率和平均服务率的历史时间序列。将得到的时间序列作为输入，对数据进行预处理，包括归一化、差分处理和滑动窗口变换，构建LSTM单元，通过记忆门机制捕捉历史请求率的时间依赖性，利用多层LSTM单元进行特征提取和时间序列预测。LSTM模块的预测结果用于更新到达率和服务率，从而估计用户请求的排队时延，使系统能更有效地应对环境的动态性，提高资源分配的准确性。最后，当智能体在状态$\boldsymbol{s}_t$执行动作$\boldsymbol{a}_t$后计算得到奖励值$R(\boldsymbol{s}_t,\boldsymbol{a}_t)$和下一个状态$\boldsymbol{s}_{t+1}$供经验回放和网络更新。在参数更新过程中，使用LSTM给出的当前任务到达率和服务率的预测值代入Kingman公式来计算排队时延。由此，学习决策层内嵌的LSTM与TD3算法协同机制学习决策层模块间输入输出映射关系如下图3所示。综上所述，学习决策层具体如下算法1所述。

算法1 基于TD3和LSTM联合的通信和计算资源分配算法

输入 观测状态$\boldsymbol{s}_t$

输出 动作$\boldsymbol{a}_t$，更新后的网络参数

（1）初始化：设置Actor网络参数$\boldsymbol{\varphi}$，Critic网络参数$\boldsymbol{\theta}_j$，目标网络参数$\boldsymbol{\varphi}'$和$\boldsymbol{\theta}_j'$，经验回放缓存D，学习率α，软更新率τ，折扣因子γ及探索噪声参数ε

（2）**for** episode $=1,2,3\cdots$**do**

（3）**for** each 时间步$=t_1,t_2,t_3\cdots$ **do**

（4） 接收观测状态$\boldsymbol{s}_t$；

（5） 使用带探索噪声的Actor网络选择动作$\boldsymbol{a}_t$；

（6） 在环境中执行动作$\boldsymbol{a}_t$；

（7） 观测下一个状态$\boldsymbol{s}_{t+1}$并计算奖励$R(\boldsymbol{s}_t,\boldsymbol{a}_t)$；

（8） 将$(\boldsymbol{s}_t,\boldsymbol{a}_t,R(\boldsymbol{s}_t,\boldsymbol{a}_t),\boldsymbol{s}_{t+1})$存入回放缓存$D$；

（9） 从D中采样小批量数据；

（10） 更新Critic网络：

（11） 计算目标动作$\mathbf{a}_{t+1}$；

（12） 根据式（26）、式（27a）、式（27b）和式（27c）计算目标Q值及Critic网络损失；

（13） 通过梯度下降更新参数$\boldsymbol{\theta}_j$；

（14） **if** 每N步更新一次Actor网络 then

（15） 根据式（27a）、式（27b）计算Actor网络损失；

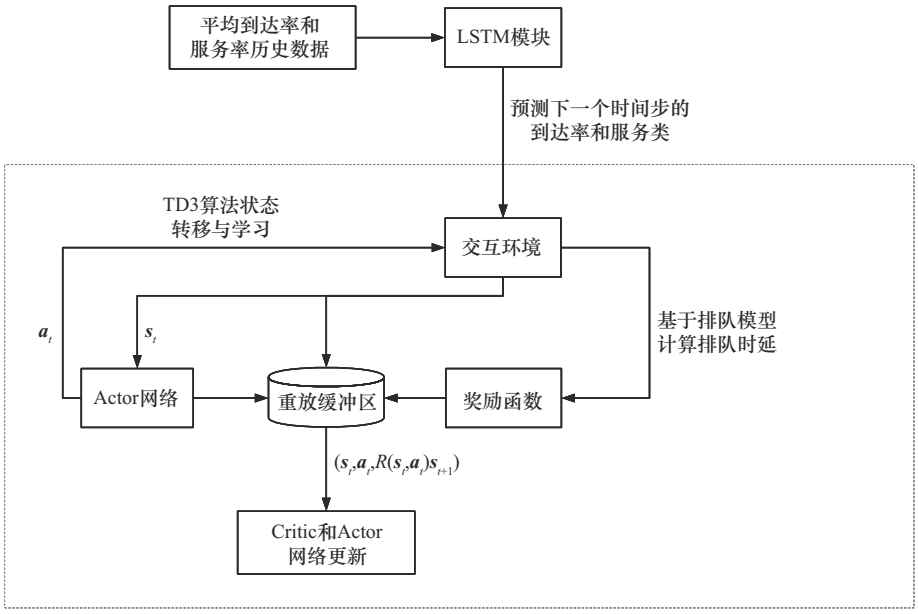


图3 学习决策层模块间输入输出映射关系

（16） 通过梯度下降更新参数$\boldsymbol{\varphi}$；

（17） 根据式（29）、式（30）软更新目标网络；

（18） **end if**

（19） 使用LSTM模块更新到达率；

（20） 记录历史到达率和服务率；

（21） 对数据进行预处理（归一化、差分、滑动窗口）；

（22） 构建并训练LSTM网络；

（23） 预测未来到达率；

（24） 更新环境参数为未来到达率；

（25） 更新状态$\boldsymbol{s}_{t+1} \leftarrow \boldsymbol{s}_t$；

（26）**end for**

（27）**end for**

（28）return $\boldsymbol{a}_t$，更新后的网络参数

3.3　优化层

高维度问题使得DRL训练优化难以收敛，而单一的传统优化算法往往会陷入局部最优。本层通过WOA确定无人机的最佳位置，并根据无人机位置的更新协助优化资源分配，以最大化网络吞吐量。WOA具有较强的全局搜索能力和较快的收敛速度，它通过初始化鲸群的位置、迭代更新位置和评估适应度函数来优化无人机的位置。为避免WOA优化对MDP假设的影响，在TD3的每一轮完整训练结束后触发WOA，在单轮训练的内部，无人机的位置被视为一个相对固定的参数，从而为TD3智能体提供了一个稳定、符合MDP假设的学习环境，WOA则利用本轮训练中收集到的全局统计信息计算适应度函数进行迭代优化得到无人机最优位置并将其用来更新环境状态空间，作为新一轮训练的初始状态，同时其他相关状态变量也相应重置，从而提高算法的整体优化效果。考虑时延和能耗都与距离有关，适配函数的参数包括伺服无人机到通信船的距离和丢包率，可由式（31）定义：

$$F_{\text{fit}} = a \cdot L_{\text{c\_s\_n}}(t) + b \cdot \text{Loss}(t) \tag{31}$$

其中，Loss(t)为伺服无人机与通信船之间的丢包率；a和b为相应的影响因子。WOA可通过螺旋更新和随机搜索策略找到无人机的最佳位置，使系统的总时延和能耗最小，其中螺旋更新和随机搜索策略可通过式（32）~式（34）计算得出：

$$\vec{\boldsymbol{X}}(t+1) = \vec{\boldsymbol{D}}' \cdot \mathrm{e}^{bl} \cdot \cos(2\pi l) + \vec{\boldsymbol{X}}^*(t) \tag{32}$$

$$\vec{\boldsymbol{D}} = |\vec{\boldsymbol{C}} \cdot \vec{\boldsymbol{X}}_{\text{rand}} - \vec{\boldsymbol{X}}(t)| \tag{33}$$

$$\vec{\boldsymbol{X}}(t+1) = \vec{\boldsymbol{X}}_{\text{rand}} - \vec{\boldsymbol{A}} \cdot \vec{\boldsymbol{D}} \tag{34}$$

其中，$\vec{\boldsymbol{D}}' = |\vec{\boldsymbol{X}}^*(t) - \vec{\boldsymbol{X}}(t)|$表示当前解与最优解之间的距离；b是定义螺旋模式的常数；l为区间[–1，1]内的数；$\vec{\boldsymbol{X}}_{\text{rand}}$表示随机选择的鲸鱼的位置向量，可以增强对稀疏节点分布的适应性，确保在拓扑突变时快速找到可行解。最后，详细的优化层如算法2所示。

算法2　基于WOA的无人机位置优化算法

输入　更新后的到达率和环境参数

输出　优化后的伺服无人机位置

（1） 初始化：设置种群大小、最大迭代次数，初始化WOA参数，并随机生成鲸群初始位置

（2） **for** episode = 1, 2, 3, ⋯ **do**

（3） 在该episode结束时，优化伺服无人机位置；

（4） **for** iteration $k = 1, 2, 3 \cdots,$ max_iterations **do**

（5） **for** i ∈ 种群 **do**

（6） 更新位置使用WOA策略：

（7） 根据式（33）执行包围猎物与随机搜索；

（8） 根据式（32）执行螺旋更新；

（9） 根据式（34）执行随机搜索；

（10） 根据式（31）计算适应度函数；

（11） 如果找到更优解则更新最佳解；

（12） **end for**

（13） **end for**

（14） 将伺服无人机位置更新为当前最佳解；

（15） 基于更新后的伺服无人机位置执行资源分配；

（16） **end for**

（17） return 伺服无人机位置

4 仿真实验及结果分析

本节将通过仿真验证提出的T-TORA算法的性能。首先，描述了仿真场景和主要参数设置。接下来，对仿真结果进行分析和讨论。

4.1 仿真设置

仿真实验构建了一个包含若干用户和边缘设备节点的支持MEC的海上通信环境，具体环境仿真主要参数设置见表2。

表2 环境仿真主要参数设置

参数	参数值
MEC服务器数量	3
用户设备数量	10
最大容忍时延	10 ms
最大信道一致性间隔	15 ms
用户请求数据量大小	[1，2] KB
计算密度	[100，125] kcycle/KB
通信船总带宽容量	[0.2，0.5] MHz
通信船总计算资源	[0.1，0.4] GHz
单次可分配带宽资源	[0.1，0.25] MHz
单次可分配计算资源	[0.05，0.2] GHz
数据卸载率	[0，1]

考虑DDPG、D4PG和TD3都是针对连续动作空间的DRL算法，同时具有相似的算法框架，为了评估基于改进TD3的T-TORA算法的性能，本文将其与这些基准算法和贪婪策略进行了比较，算法仿真主要参数设置见表3，展示了T-TORA算法的具体参数。基准算法的基础超参数设置与T-TORA算法一致。

表3 算法仿真主要参数设置

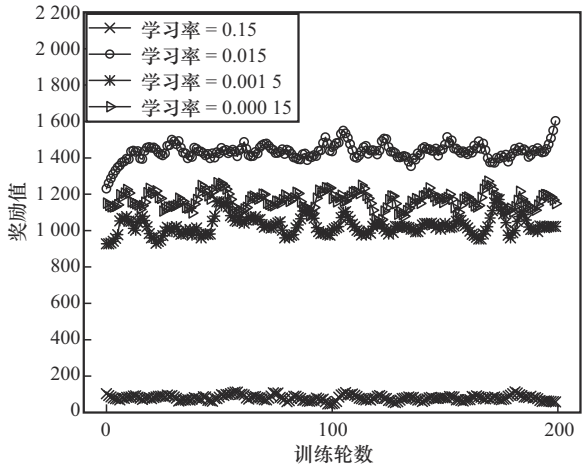
参数	参数值	参数	参数值
TD3算法训练轮数	200	LSTM激活函数	tanh
LSTM输入序列长度	5	WOA螺旋常数	1
WOA种群个体数	50	TD3策略噪声标准差	0.5
TD3算法学习率	0.015	LSTM学习率	0.001
LSTM特征数	2	WOA鲸鱼初始化半径	45
WOA最大迭代次数	100	TD3重放缓存区大小	12 000
TD3算法折扣因子	0.99	LSTM训练轮数	200
LSTM单元数	50	TD3算法预热轮数	50
WOA搜索空间维度	3	LSTM批次大小	32
TD3算法软更新率	0.005	TD3算法批次大小	32

4.2 T-TORA算法参数评估

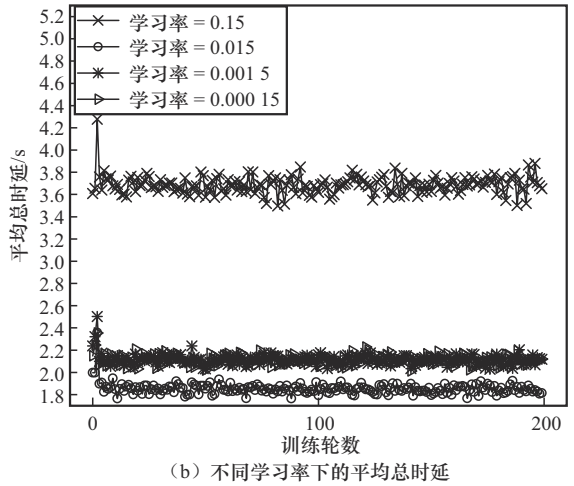
T-TORA算法在不同学习率下的性能指标如图4所示。其中，图4（a）表示不同学习率下的奖励对比，为奖励值曲线；图4（b）表示不同学习率下的平均总时延，为平均总时延对比；图4（c）表示不同学习率下的平均总能耗，为平均总能耗对比。由图4可得，在学习率为0.015时，提出的T-TORA算法有最低的平均总延时、最低的平均总能耗和最高的奖励值。这是由于合适的学习率可以有效优化目标函数，使模型更快地找到全局最优点，而过高的学习率可能会导致模型震荡无法收敛，过低的学习率则可能使收敛速度严重下降。由此可得出结论，当学习率为0.015时，T-TORA算法具有更优的性能表现。

下面针对T-TORA算法在不同到达率下的奖励性能和收敛性展开评估。对整体系统而言，到达率越高，负荷越大。不同服务请求到达率下的收敛性能如图5所示，评估到达率为10、20

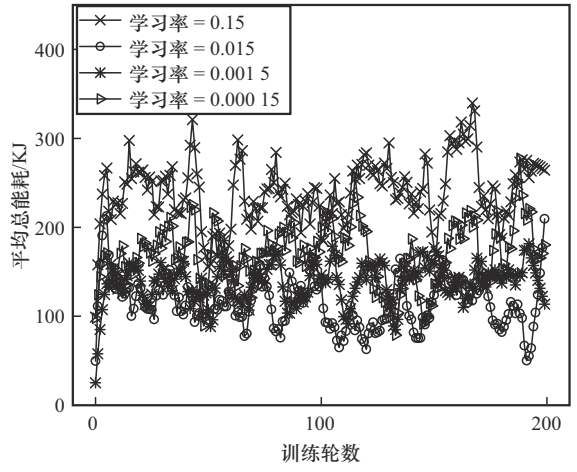
和30的情况，算法在大约25个回合时收敛，随着到达率的增加，算法奖励值也随之增加并最终收敛。



（a）不同学习率下的奖励对比



（b）不同学习率下的平均总时延



（c）不同学习率下的平均总能耗

图4　T-TORA 算法在不同学习率下的性能指标

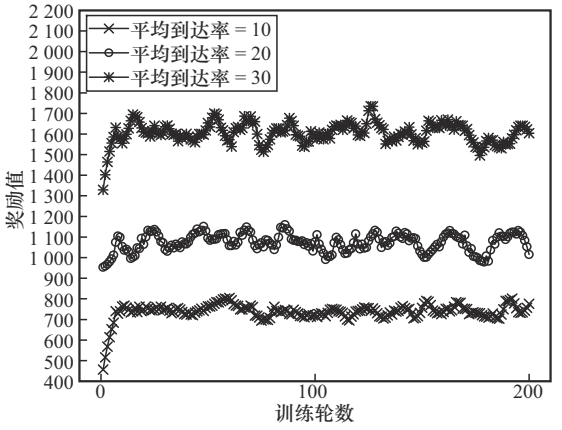


图5　不同服务请求到达率下的收敛性能

4.3　算法奖励值对比

下面为验证所提算法的有效性，不同算法的奖励值对比如图6所示，对比了T-TORA算法和其他基准算法在测试环境下的奖励值。由图6可得，T-TORA通过联合优化资源分配与任务调度，实现了更快的收敛速度，其训练初期出现的适度波动反映了持续的寻优过程，最终累计回报最高。对比基线中，TD3算法的双Critic网络和时延更新机制有效提高了策略的稳定性，减少了Q值被高估的问题，表现出更高更稳定的奖励趋势。DDPG算法在复杂环境下容易出现Q值高估与训练不稳定的情况，导致奖励偏低且波动较大。D4PG算法在环境变化时策略调整滞后，致使学习效率受限，难以在有限训练轮数内获得高质量的分配策略。最后，由于策略刚性与对环境动态的适应性不足，贪婪策略的奖励值最低。

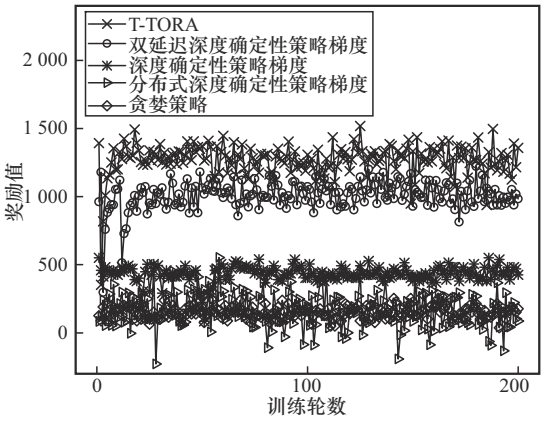


图6　不同算法的奖励值对比

4.4 T-TORA算法性能评估

在训练中，代理不断更新策略，以实现长期回报的最大化，这意味着更低的时延和能耗，即更高的网络吞吐量。不同服务请求到达率下的网络吞吐量对比如图7所示，分别展示了不同任务请求到达率下T-TORA算法和其他基准算法在网络吞吐量方面的性能。随着到达率的增加，系统网络吞吐量的总体趋势也在增加，相比之下，本文所提出的T-TORA算法更稳定，性能更优。DDPG算法存在高估问题，导致整体训练结果出现波动，性能较差。TD3算法虽有效缓解了DDPG的高估缺陷，但未充分考虑无人机位置部署对系统整体性能的影响，未能准确建模等待状态中的时延特性，导致其网络吞吐量仍低于T-TORA算法。D4PG算法的分布式特性使其在资源受限条件下收敛速度较慢，难以在高到达率环境中快速适应。随着到达率的增加，贪婪策略的吞吐量呈上升趋势，但由于其策略固定，性能最差。

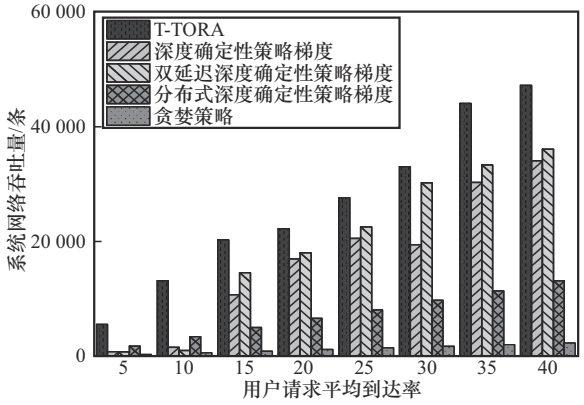


图7 不同服务请求到达率下的网络吞吐量对比

不同带宽资源下的网络吞吐量对比如图8所示，分别展示了不同带宽资源容量条件下，所提出的T-TORA算法与其他基准算法的网络吞吐量对比。由图8可得，系统的网络吞吐量随着带宽资源容量的增大而增大，这是因为当设备的带宽资源容量增大时，数据传输速率增大，传输端的数据传输时延缩短，即可在时延限制下传输更多数据，网络吞吐量随之增大。相比之下，本文提出的T-TORA算法在平滑度和吞吐量方面表现更优，这是因为T-TORA算法包含了TD3算法波动小的优点，而且与TD3算法相比，考虑了等待时延的计算和最优位置的影响，使整个算法对环境的动态变化有了更全面的考虑。D4PG算法无法有效利用带宽资源的扩展，导致其无法充分利用新增带宽来提高吞吐量，策略更新不够灵活。贪婪策略在不同带宽资源下的吞吐量最低，这是因为其策略刚性无法适应高度变化的动态环境。

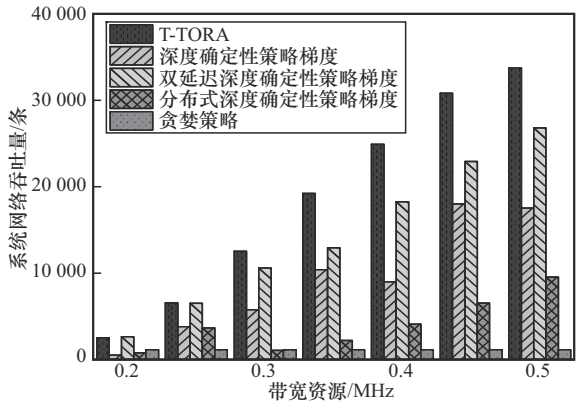


图8 不同带宽资源下的网络吞吐量对比

不同计算资源下的网络吞吐量对比如图9所示，分别展示了不同计算资源容限条件下，T-TORA算法与其他基准算法在网络吞吐量方面的性能。如图所示，系统网络吞吐量的总体趋势是随着计算资源容限的增加而增加，这是因为当设备的计算资源容限增加时，设备对数据请求的计算速率上升，在计算端对数据进行计算处理的时延降低，即更多的数据在时延限制下完成了计算处理，网络吞吐量随之增加。相比之下，本文提出的T-TORA算法在平滑度和收敛速度方面更优，验证了T-TORA算法对动态变化的环境具有更好的适应性。DDPG算法由于高估问题，存在波动且性能较差。TD3算法上升趋势稳定，但缺乏全局考量，性能相比T-TORA算法较差。D4PG算法的分布式架构不适合处理计算资源的动态变化，尤其是资源不足和资源过剩状态之间的灵活调度。同时，在不同计算资源下，贪婪策略的固定策略导致性能最差。

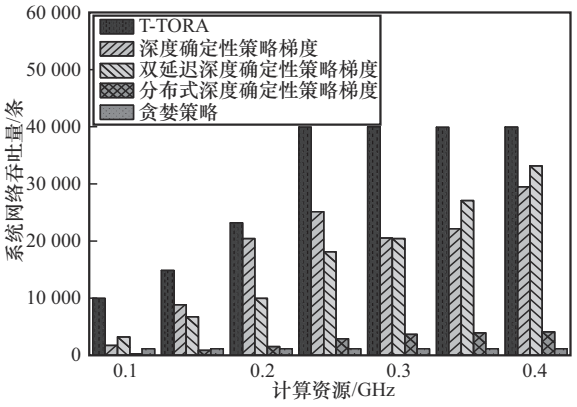


图9　不同计算资源下的网络吞吐量对比

下面，为验证所提算法在不同网络规模下的有效性，不同用户数量下的网络吞吐量对比如图10所示，展示了固定5架无人机时，不同用户数量下T-TORA算法和其他基准算法的平均网络吞吐量性能。由图10可得，系统平均吞吐量随用户数增加先升后降，这是因为用户数量增多导致请求总量提升，但固定资源下系统负载逐渐饱和，使得吞吐量达到瓶颈后下降。如图所示，当用户数达到30时系统负载饱和，相比之下，T-TORA在稳定性与性能方面均优于对比算法，更适应动态规模的网络环境。DDPG算法因过高估计问题导致波动大且性能差，TD3算法相比DDPG更加稳定，但全局环境适应能力不足。D4PG算法因计算复杂度和空间复杂度高，交互效率低，性能表现一般。贪婪策略则因无法动态调整，性能无法优化改善且随用户数增加而下降。

最后，不同干扰强度下的性能对比如图11所示，展示了不同干扰强度下不同算法的性能对比，其中，图11（a）表示不同干扰强度下的系统总时延对比，图11（b）表示不同干扰强度下的系统网络吞吐量对比。由图11（a）可得，随着干扰方差增大，系统平均总时延总体呈上升趋势。根据香农公式，干扰强度增大，数据传输速率逐渐降低，导致数据传输时延增加，使得系统平均总时延逐步上升。相比之下，本文提出的T-TORA实现了更低的上升速率和更低的系统平均总时延，由此验证了算法在高干扰情况下的有效性。同时，从图11（b）可以看出，随着干扰方差增大，系统网络吞吐量在逐步下降，相比之下，本文提出的T-TORA算法实现了更低的下降速率和更高的系统网络吞吐量，进一步验证了算法对于高干扰环境的适应性和有效性。

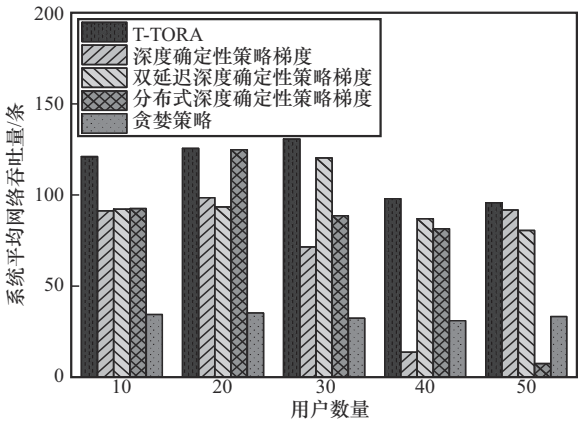


图10　不同用户数量下的网络吞吐量对比

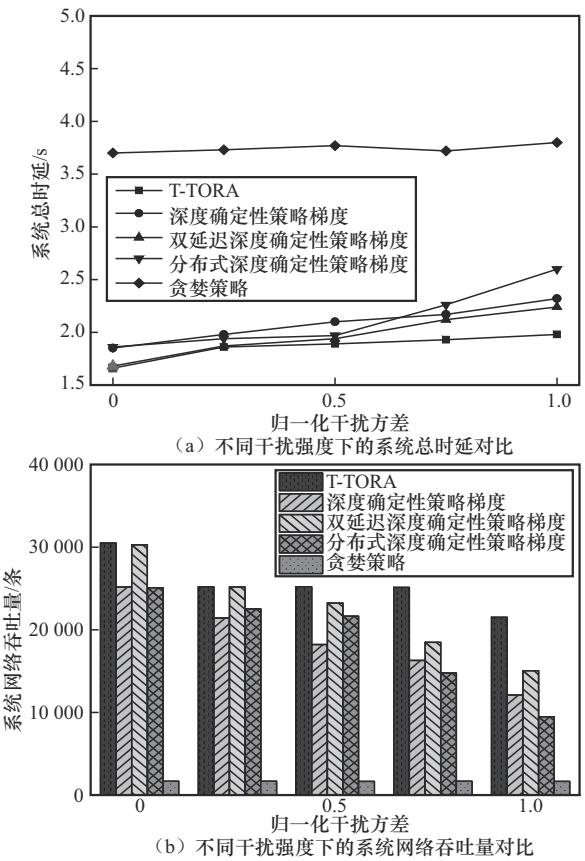


图11　不同干扰强度下的性能对比

综上所述，在不同网络环境下，提出的T-TORA算法在网络吞吐量方面，性能显著优于其他基准算法，通过上述对比实验验证了所提出算法的有效性和可行性。

4.5 算法复杂度对比

下面从时间复杂度和空间复杂度对提出的T-TORA算法和4种基准算法在部署时的计算和存储开销进行评估。在时间复杂度方面，T-TORA算法的计算开销主要包括TD3的Actor-Critic网络前向传播、LSTM预测和WOA位置优化。具体细节如下：

$$T_{\mathrm{TD3}}=O\left(\sum_{j=1}^{J}u_{a,j}\cdot u_{a,j+1}+2\sum_{f=1}^{F}u_{c,f}\cdot u_{c,f+1}\right) \tag{35}$$

$$T_{\mathrm{LSTM}}=O(T\cdot N_l^2) \tag{36}$$

$$T_{\mathrm{WOA}}=O(K\cdot P\cdot D) \tag{37}$$

其中，$u_{a,j}$表示Actor网络第j层的神经元数量；$u_{c,f}$是Critic网络第f层的神经元数量。LSTM的时间复杂度取决于时间步长和隐藏单元的数量，在式（36）中，T为时间步长，N_l代表隐藏单元的数量。最后，在WOA中，每次迭代都需要计算种群中P个个体的适应度，因此WOA优化的时间复杂度可以定义为式（37），其中，K为最大迭代次数，P为种群内个体数量，D为向量维度。综上所述，提出的T-TORA算法的时间复杂度T_{total}计算式如下：

$$T_{\mathrm{total}}=T_{\mathrm{TD3}}+T_{\mathrm{LSTM}}+T_{\mathrm{WOA}} \tag{38}$$

考虑4种基准算法，首先，DDPG的时间复杂度仅包含单个Critic网络和Actor网络的前向和反向传播，具体可由下式定义：

$$T_{\mathrm{DDPG}}=O\left(\sum_{j=1}^{J}u_{a,j}\cdot u_{a,j+1}+\sum_{f=1}^{F}u_{c,f}\cdot u_{c,f+1}\right) \tag{39}$$

其中，参数定义与式（35）相同。TD3的时间复杂度定义与式（35）相同。而D4PG在DDPG的基础上采用分布式并行训练，具体由式（40）定义：

$$T_{\mathrm{D4PG}}=O\left(W\left[\sum_{j=1}^{J}u_{a,j}\cdot u_{a,j+1}+\sum_{f=1}^{F}u_{c,f}\cdot u_{c,f+1}\right]\right) \tag{40}$$

其中，W为并行的Worker数量。贪婪策略无须学习，仅基于规则决策，时间复杂度计算式如下：

$$T_{\mathrm{Greedy}}=O(1) \tag{41}$$

综上，在时间复杂度方面，贪婪策略复杂度最低且固定，但无法适应动态环境；DDPG时间复杂度较低但策略稳定性不足；TD3因采用双Critic网络，其复杂度高于DDPG，但有效提升了策略稳定性；D4PG因分布式架构引入并行Worker，时间复杂度线性增加；提出的T-TORA算法在TD3基础上引入LSTM和WOA模块，虽然增加了计算开销，但通信代价低于D4PG，且在策略稳定性和优化性能上优于其他基准算法。为明确T-TORA算法中LSTM和WOA对整体算法时间复杂度的贡献比例，进一步分析T-TORA算法各模块的调用频率和单次执行复杂度。首先，TD3算法的Actor-Critic网络更新为算法的核心决策过程，每一个时间步均需更新网络，是高频调用但单次复杂度中等的模块。其次，LSTM用于预测用户请求到达率和服务率，需要在每一个时间步执行以确保对环境动态的实时捕捉，同样是高频调用但单次复杂度较低的模块。最后，WOA用于优化无人机位置，每轮训练结束时调用一次，是低频调用但单次复杂度较高的模块。3个模块的执行频率不同，因此，在一个完整的训练回合中，其总耗时计算式如下：

$$T_{\mathrm{episode}}=T_{\mathrm{step}}\cdot(T_{\mathrm{TD3}}+T_{\mathrm{LSTM}})+T_{\mathrm{WOA}} \tag{42}$$

其中，T_{step}为一个完整训练回合所包含的时间步。由此可得LSTM和WOA对总耗时的贡献比例，如下所示：

$$R_{\mathrm{contri\_LSTM}}=\frac{T_{\mathrm{step}}\cdot T_{\mathrm{LSTM}}}{T_{\mathrm{episode}}} \tag{43}$$

$$R_{\mathrm{contri\_WOA}}=\frac{T_{\mathrm{WOA}}}{T_{\mathrm{episode}}} \tag{44}$$

由式（43）可得，LSTM 的时间复杂度开销线性稳定；而式（44）表明 WOA 的时间复杂度开销固定，对总耗时的贡献比例随训练轮数的增加而降低。为综合分析算法性能与算法时间复杂度间的权衡，基于仿真结果和上述分析可得算法平均奖励值与算法单步决策计算开销的权衡曲线，系统平均网络吞吐量与单步耗时权衡如图 12 所示。

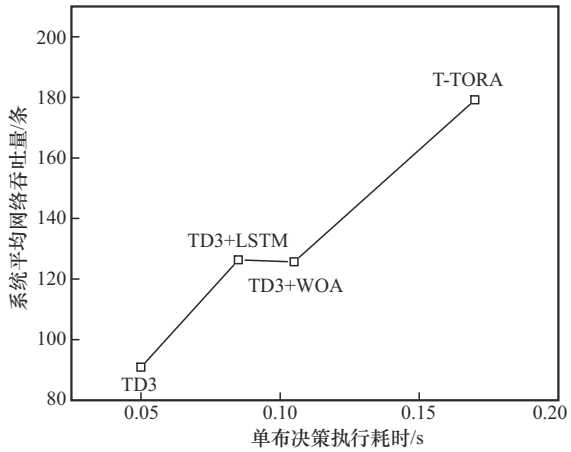


图 12　系统平均网络吞吐量与单步耗时权衡

在空间复杂度方面，提出的 T-TORA 算法空间复杂度主要由神经网络参数存储、经验重放缓冲区和种群存储 3 部分组成，具体如下所示：

$$G_{\mathrm{T-TORA}}=O\left(\sum_j u_{a,j}^2+2\sum_j u_{c,j}^2+N_l^2\right) \quad (45)$$

$$S_{\mathrm{Buffer}}=O(D\cdot(d_s+d_a)) \quad (46)$$

$$S_{\mathrm{WOA}}=O(P\cdot(D+1)) \quad (47)$$

T-TORA 算法的神经网络参数总数可由式（45）计算，其中，$\sum_j u_{a,j}^2$ 为 Actor 网络参数总数，$2\sum_j u_{c,j}^2$ 为 Critic 网络参数总数，N_l^2 为 LSTM 模块参数总数。经验重放缓冲区的空间复杂度由式（46）定义，其中，D 表示存储容量，d_s 是状态空间的维度，d_a 是动作空间的维度。WOA 需要存储 P 个个体的位置和适应度值，因此，WOA 种群存储的空间复杂度可由式（47）定义。由此，所提出的 T-TORA 算法的空间复杂度 $S_{\mathrm{T-TORA}}$ 可由式（48）定义：

$$S_{\mathrm{T-TORA}}=G_{\mathrm{T-TORA}}+S_{\mathrm{Buffer}}+S_{\mathrm{WOA}} \quad (48)$$

下面考虑 4 种基准算法。首先，DDPG 的空间复杂度主要包含网络参数存储和经验重放缓存区，具体由式（49）定义：

$$G_{\mathrm{DDPG}}=O\left(\sum_j u_{a,j}^2+\sum_j u_{c,j}^2\right) \quad (49)$$

其中，参数定义与式（45）和式（46）相同，由此可得 DDPG 总的空间复杂度如下：

$$S_{\mathrm{DDPG}}=G_{\mathrm{DDPG}}+S_{\mathrm{Buffer}} \quad (50)$$

其次，TD3 相比 DDPG 增加了 1 个 Critic 网络，因此其总的空间复杂度可由式（51）定义：

$$S_{\mathrm{TD3}}=O\left(\sum_j u_{a,j}^2+2\sum_j u_{c,j}^2\right)+O(D(d_s+d_a)) \quad (51)$$

其中，参数定义与式（45）和式（46）相同。D4PG 则是引入了 W 个并行的 Worker，每个 Worker 均有一套 Actor-Critic 网络，由此可得 D4PG 总的空间复杂度计算式如下：

$$S_{\mathrm{D4PG}}=O\left(W\left(\sum_j u_{a,j}^2+2\sum_j u_{c,j}^2\right))+O(WD(d_s+d_a)\right) \quad (52)$$

最后，贪婪策略不存储模型和回放区，内存开销最低，总的空间复杂度计算式如下：

$$S_{\mathrm{Greedy}}=O(1) \quad (53)$$

综上所述，贪婪策略空间复杂度最低，但无法适应动态海上通信环境下的决策需求；DDPG 相比 TD3 的存储开销更少，但策略稳定性较差；D4PG 则由于并行机制导致内存开销线性增加；提出的 T-TORA 算法相比 TD3 增加了 LSTM 和 WOA 种群的存储，通过增加部分空间开销换取更加稳定的策略优化和性能提升。

5　结束语

本文研究了动态海事 MEC 网络架构中联合资源分配和任务卸载策略的优化问题。目标是在时延和能耗的约束下优化最大网络吞吐量。为此，提出了 T-TORA 算法，该算法利用改进的

TD3方法来联合优化边缘节点的任务卸载和资源分配。T-TORA通过在线学习适应动态变化的条件，并通过确定伺服无人机的最佳定位来优化资源分配。仿真结果表明，与基线算法相比，T-TORA能显著提高网络吞吐量。

参考文献：

[1] LIU S L, ZHU L J, HUANG F H, et al. A survey on air-to-sea integrated maritime Internet of Things: enabling technologies, applications, and future challenges[J]. Journal of Marine Science and Engineering, 2024, 12(1): 11.

[2] AKHTAR M W, SAEED N. UAVs-enabled maritime communications: UAVs-enabled maritime communications: opportunities and challenges[J]. IEEE Systems, Man, and Cybernetics Magazine, 2023, 9(3): 2-8.

[3] SHIRIN ABKENAR F, RAMEZANI P, IRANMANESH S, et al. A survey on mobility of edge computing networks in IoT: state-of-the-art, architectures, and challenges[J]. IEEE Communications Surveys & Tutorials, 2022, 24(4): 2329-2365.

[4] QIN Z, HE S S, WANG H, et al. Air-ground collaborative mobile edge computing: Architecture, challenges, and opportunities[J]. China Communications, 2024, 21(5): 1-16.

[5] QIU Y, NIU J W, ZHU X Z, et al. Mobile edge computing in space-air-ground integrated networks: architectures, key technologies and challenges[J]. Journal of Sensor and Actuator Networks, 2022, 11(4): 57.

[6] NING Z L, HU H, WANG X J, et al. Mobile edge computing and machine learning in the Internet of unmanned aerial vehicles: a survey[J]. ACM Computing Surveys, 2024, 56(1): 1-31.

[7] SHARMA A, DIWAKER C, NADIYAN M. Analysis of offloading computation in mobile edge computing (MEC): a survey[C]//Proceedings of the 2022 Seventh International Conference on Parallel, Distributed and Grid Computing (PDGC). Piscataway: IEEE Press, 2022: 280-285.

[8] DJIGAL H, XU J, LIU L F, et al. Machine and deep learning for resource allocation in multi-access edge computing: a survey[J]. IEEE Communications Surveys & Tutorials, 2022, 24(4): 2449-2494.

[9] ADIL M, SONG H B, MASTORAKIS S, et al. UAV-assisted IoT applications, cybersecurity threats, AI-enabled solutions, open challenges with future research directions[J]. IEEE Transactions on Intelligent Vehicles, 2024, 9(4): 4583-4605.

[10] ZHANG P Y, WANG C, JIANG C X, et al. UAV-assisted multi-access edge computing: technologies and challenges[J]. IEEE Internet of Things Magazine, 2021, 4(4): 12-17.

[11] LIU Z W, CAO Y, GAO P, et al. Multi-UAV network assisted intelligent edge computing: challenges and opportunities[J]. China Communications, 2022, 19(3): 258-278.

[12] NOMIKOS N, GKONIS P K, BITHAS P S, et al. A survey on UAV-aided maritime communications: deployment considerations, applications, and future challenges[J]. IEEE Open Journal of the Communications Society, 2022(4): 56-78.

[13] KIM M, JANG J, CHOI Y, et al. Distributed task offloading and resource allocation for latency minimization in mobile edge computing networks[J]. IEEE Transactions on Mobile Computing, 2024, 23(12): 15149-15166.

[14] QIAN L P, SHI B H, WU Y, et al. NOMA-enabled mobile edge computing for Internet of Things via joint communication and computation resource allocations[J]. IEEE Internet of Things Journal, 2020, 7(1): 718-733.

[15] WANG Y, TAO X F, HOU Y T, et al. Effective capacity-based resource allocation in mobile edge computing with two-stage tandem queues[J]. IEEE Transactions on Communications, 2019, 67(9): 6221-6233.

[16] XING H, LIU L, XU J, et al. Joint task assignment and resource allocation for D2D-enabled mobile-edge computing[J]. IEEE Transactions on Communications, 2019, 67(6): 4193-4207.

[17] TUN Y K, DANG T N, KIM K, et al. Collaboration in the sky: a distributed framework for task offloading and resource allocation in multi-access edge computing[J]. IEEE Internet of Things Journal, 2022, 9(23): 24221-24235.

[18] EI N N, YOON J S, HONG C S. Energy-aware task offloading and resource allocation in space-aerial-integrated MEC system[C]// Proceedings of the 2022 23rd Asia-Pacific Network Operations and Management Symposium (APNOMS). Piscataway: IEEE Press, 2022: 1-6.

[19] AN X M, FAN R F, HU H, et al. Joint task offloading and resource allocation for IoT edge computing with sequential task dependency[J]. IEEE Internet of Things Journal, 2022, 9(17): 16546-16561.

[20] WU J, JIA M, GUO Q, et al. Joint optimization computation offloading and resource allocation for LEO satellite with edge computing[C]//Proceedings of the 2023 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB). Piscataway: IEEE Press, 2023: 1-5.

[21] GUO F X, YU F R, ZHANG H L, et al. Adaptive resource allo-

cation in future wireless networks with blockchain and mobile edge computing[J]. IEEE Transactions on Wireless Communications, 2019, 19(3): 1689-1703.

[22] NATH S, WU J X. Deep reinforcement learning for dynamic computation offloading and resource allocation in cache-assisted mobile edge computing systems[J]. Intelligent and Converged Networks, 2020, 1(2): 181-198.

[23] LIANG Y, SUN H F. Optimizing task processing efficiency in MEC networks through cooperative offloading and resource allocation[C]//Proceedings of the 2024 6th International Conference on Communications, Information System and Computer Engineering (CISCE). Piscataway: IEEE Press, 2024: 296-301.

[24] WEI Z, HE R X, LI Y N. Deep reinforcement learning based task offloading and resource allocation for MEC-enabled IoT networks[C]//Proceedings of the 2023 IEEE/CIC International Conference on Communications in China (ICCC Workshops). Piscataway: IEEE Press, 2023: 1-6.

[25] YU L, JIANG S R, ZHENG J, et al. A DQN-based joint computing offloading and resource allocation algorithm for MEC networks[C]//Proceedings of the ICC 2023 - IEEE International Conference on Communications. Piscataway: IEEE Press, 2023: 2553-2558.

[26] ZHANG B Y, JIANG Y X, HUANG Y G, et al. A DRL scheme for resource allocation in the MEC-empowered CF-mMIMO system[C]//Proceedings of the 2023 IEEE 23rd International Conference on Communication Technology (ICCT). Piscataway: IEEE Press, 2023: 495-500.

[27] HAZARIKA B, SINGH K, BISWAS S, et al. DRL-based resource allocation for computation offloading in IoV networks[J]. IEEE Transactions on Industrial Informatics, 2022, 18(11): 8027-8038.

[28] LIU Y, YU H M, XIE S L, et al. Deep reinforcement learning for offloading and resource allocation in vehicle edge computing and networks[J]. IEEE Transactions on Vehicular Technology, 2019, 68(11): 11158-11168.

[29] Kingman J F C. The single server queue in heavy traffic[C]//Mathematical Proceedings of the Cambridge Philosophical Society. Cambridge University Press, 1961, 57(4): 902-904.

[作者简介]

徐艳丽（1984– ），女，上海海事大学信息工程学院副院长、教授、博士生导师，主要研究方向为海事通信、边缘计算和自动驾驶等。

周子睿（2001– ），男，上海海事大学信息工程学院硕士生，主要研究方向为海事通信和边缘计算。