基于MADRL的海上物联网任务卸载优化方案

摘　要：为解决海上网络覆盖范围小、计算能力弱的问题，引入了空天地海一体化网络场景下的任务卸载决策。考虑计算任务的卸载成功率、能耗约束、近海场景与远海场景的环境差异以及空天地海一体化场景的动态性，构建了一种适用于空天地海一体化网络的任务卸载架构，提出了一种基于深度强化学习的多智能体协同任务卸载方案。实验结果表明，相较于基于 MADQN 算法的卸载方案、基于 DDPG 算法的卸载方案和随机策略的卸载方案，所提方案在卸载成功率方面分别提高 5.08%、21.71% 和 60.48%，在时延方面分别降低 11.65%、18.64% 和 64.60%，在能耗方面分别降低 11.57%、9.66% 和 10.38%。

关键词：空天地海一体化网络；任务卸载；多智能体；深度强化学习

doi：10.11959/j.issn.2096−3750.XXXX.

Research on Task Offloading Optimization Scheme Based on MADRL for Maritime IoT

Abstract: To address the issues of limited network coverage and weak computing capabilities in maritime environments, a task offloading scheme in the space-air-ground-sea integrated network scenario was introduced. Considering factors such as the offloading success rate of computational tasks, energy consumption constraints, environmental differences between near-shore and offshore scenarios, and the dynamic nature of the integrated space-air-ground-sea environment, a task offloading framework suitable for space-air-ground-sea integrated networks was constructed. A multi-agent collaborative task offloading scheme based on deep reinforcement learning was proposed. Experimental results demonstrate that, compared to offloading schemes based on the MADQN algorithm, the DDPG algorithm, and random policy, the proposed scheme improves the offloading success rate by 3.09%, 18.42%, and 66.42%, respectively, reduces delay by 19.07%, 21.53%, and 65.02%, respectively, and lowers energy consumption by 10.59%, 8.20%, and 8.75%, respectively.

Key words: Space-Air-Ground-Sea Integrated Networks, Task Offloading, Multi Agent, Deep Reinforcement Learning

0　引 言

随着全球化的加速和国际贸易的增加，海事活动的重要性日益凸显。在这一背景下，船舶在航次中需保持安全、稳定的通信，并需要研究人员开发海上物联网(M-IOT, Maritime Internet-of-Things)应用，例如智能港口和自主导航[1]。这些M-IOT应用通常需要大量计算与存储资源来保证服务质量(QoS, Quality-of-Service)。但由于海上移动设备的计算架构与资源储备不足，这类应用的高资源需求较难得到满足。针对这一问题，移动边缘计算(MEC, Mobile Edge Computing)作为一种新颖有效的方案，被广泛使用。在MEC中，移动设备可以将任务卸载到附近的MEC服务器中，这类服务器具有丰富计算资源和计算能力。完成计算后，服务器将结果返回给用户，从而更好地支持复杂任务的计算需求。与传统云计算相比，MEC服务器部署在更靠近用户的位置，显著降低了任务传输时延，有效地提高了QoS。

然而，由于海上用户移动性、海上基础设施匮乏以及远海领域设施维护成本高昂、难度较大，MEC服务器的部署面临巨大挑战。随着卫星通讯技术的发展，一种新兴的空天地海一体化网络架构(SAGSIN, Space-Air-Ground-Sea Integrated Network)引起了研究者的关注。SAGSIN在海基网络的基础上，协同空基、天基和地基网络，可为海上用户提供更全面且灵活的网络接入。在SAGSIN中，地基网络主要包括地面基站(BS, Base Station)，能够实现对近海区域的广泛覆盖，为海上提供稳定可靠的

收稿日期：XXXX−XX−XX；修回日期：XXXX−XX−XX

网络接入服务；海基网络包括海上浮标等。在远离地面的海上场景中，海上浮标等设施的部署难度大、后期维护成本高，因此海基网络也主要服务于靠近地面的近海区域。海基网络与地基网络协同工作，能为近海区域提供更全面的网络覆盖，从而提升网络可靠性和稳定性。而天基网络由不同轨道的通信卫星构成，主要包括地球静止轨道(GEO, Geostationary Earth Orbit)卫星、中轨道(MEO, Medium Earth Orbit)卫星和低轨道(LEO, Low Earth Orbit)卫星，具备全球覆盖、泛在连接的优势。空基网络由空中飞行设备(如无人机(UAV, Unmanned Aerial Vehicle)、低空通信平台(LAP, Low Altitude Platforms))组成，用于提供无线通信，其优势为覆盖范围大、部署灵活。天基网络与空基网络协同，共同为海上区域提供更全面的网络覆盖。

在真实SAGSIN场景里，移动设备任务量与任务大小一般会随时间变化。如何针对动态环境设计出高效的任务卸载策略，成为一个亟待深入研究的问题。现有研究工作多采用迭代算法，然而这类算法难以适应动态化的SAGSIN环境。在机器学习领域，新兴的深度强化学习展现出了巨大潜力，为解决复杂问题开辟了新方向。

针对上述问题，本文提出了一种基于多智能体深度强化学习(MADRL, Multi-Agent Deep Reinforcement Learning)的协同任务卸载方案。本文的主要贡献如下：

(1) 所提出的SAGSIN系统充分考虑了近海场景与远海场景的区别，将优化目标定为最大化任务卸载成功率、最小化能耗，并将卸载问题分为两个子问题加以解决。

(2) 设计了一种基于多智能体深度确定性策略梯度算法(MADDPG, Multi-Agent Deep Deterministic Policy Gradient)的协同任务卸载方案，借助深度强化学习，智能体可在动态环境下做出最优卸载决策。

(3) 设计了一种基于深度Q网络(DQN, Deep Q Network)算法的任务卸载算法方案，使近海场景中接收智能体传输任务的基站，能做出将任务卸载至海上浮标的最优决策。

1　相关工作

MEC在车联网场景下已得到广泛应用。Li等[2]研究了车辆边缘计算环境下，如何通过任务卸载满足计算密集型与时延敏感型车辆应用对高计算能力的需求。Liu等[3]提出了一种智能驱动的车载边缘计算网络架构，设计了任务卸载与服务缓存的联合优化机制，通过异步分布式强化学习算法实现最优卸载决策与资源管理。Zhang等[4]研究了基于多接入边缘计算的车联网环境，提出了一种基于深度确定性策略梯度的负载均衡算法，用于解决任务卸载与资源分配问题。现有研究多聚焦于地面网络的任务卸载。并且当大量的任务在较短的时间内卸载到同一边缘服务器，该服务器可能过载，导致任务中断甚至失败。另一方面，在基础设施匮乏的偏远地区，边缘服务器的缺失会进一步加剧计算资源的不足。

为扩大网络覆盖范围，部分学者引入了无人机辅助计算卸载。Hevesli等[5]针对无人机在能源和覆盖范围方面的资源限制，提出了带Beta分布的多智能体近端策略优化，通过优化无人机轨迹、计算资源分配及队列感知的任务卸载决策，实现了整体能耗最小化。Subhrajit等[6]通过结合无人机与地面边缘服务器的能力，最大化用户满意度，提升了服务提供商的利润。Chen等[7]提出了一种多无人机辅助车联网的新型资源分配和协同卸载框架，有效解决了无人机辅助车联网系统中的资源分配与计算卸载问题，显著提升了系统性能及通信覆盖范围。

部分学者通过卫星实现了更广阔的网络覆盖，以支持计算卸载。Chai等[8]提出了基于合作斯塔克伯格博弈的计算卸载算法，构建了集成卫星-地面的车辆网络环境，解决了6G边缘网络中车辆计算任务卸载问题。Lan等[9]提出了一种面向集成卫星-地面网络且考虑安全性的任务卸载方案，借助卫星计算资源，为地面更高效、安全的卸载服务。Li等[10]量化了任务执行中的时延与能耗，提出了具有非线性收敛因子和自适应权重的离散鲸鱼优化算法。该算法通过优化卸载决策、功率控制等方面，实现了系统成本函数的最小化。然而，上述研究仅聚焦空-地或天-地网络协同，未能充分整合三类网络资源。

空天地一体化网络(SAGIN, Space-Air-Ground Integrated Network)通过整合空基、天基与地基网络，实现了三者的协同而备受关注。Zeng等[11]在空天地一体化车载网络中提出了基于深度强化学习的

计算卸载与资源分配策略，可显著提高任务成功率并降低任务时延。Li等[12]研究了SAGIN支持车联网服务的场景，提出了动态虚拟网络功能映射与调度的解决方案，实现了服务提供商利润最大化及QoS保障。Xie等[13]利用SAGIN覆盖范围广的特点，设计了辅助车辆计算的集成网络框架，旨在提升卫星及UAV的利用率。在保证时延和公平性效用最大化的前提下，为偏远地区的车联网服务提供支持。在此基础上拓展海基网络的方向逐渐受到更多研究者的关注。Lin等[14]提出了一种双重迭代惩罚限制模拟退火的粒子群优化算法，解决了SAGSIN中船舶计算卸载与带宽分配的非凸非线性能耗最小化问题。Xu等[15]在SAGSIN场景中，借助DQN算法解决了海上网络的联合通信与计算资源分配问题，优化了计算任务资源利用及执行时延。Meng等[16]提出了适应时变网络环境的联合感知-通信-计算-驱动框架及信息同步成本指标，有效提升了SAGSIN下任务关键型应用的性能。然而，现有研究对SAGSIN的考虑不够全面。例如，研究仅关注近海或远海场景或未充分考虑两者差异及海上基础设施的计算能力。

针对现有研究的不足与挑战，本文设计了一种适用于SAGSIN的任务卸载架构。为实现海上场景中的最优卸载策略，本文提出了一种基于MADRL的协同任务卸载方案。分析并综合考虑近远海场景的区别后，将卸载问题分为船舶卸载问题和基站卸载问题两个子问题，并使用深度强化学习算法求解。

2　系统模型

本文提出一种适用于SAGSIN的任务卸载架构，由太空、天空、地面、海洋层组成，如图1所示。太空层：卫星作为助手，为近海和远海的船舶服务。天空层：一批具有通信能力和计算资源的UAV分布在海拔H处，其分布遵循二维泊松点过程，密度为λ_{U}[17]。地面层：靠近海岸的BS。船舶与BS之间距离满足$d_{V2B}\in(\mathrm{r},R)$，其中r为BS到船舶的安全距离，R为船舶通信距离。海洋层：海岸边分布带有MEC服务器的浮标。BS到海上浮标的距离满足$d_{B2b}\in(\mathrm{r},R_b)$，其中$R_b$为浮标距离BS最大距离。

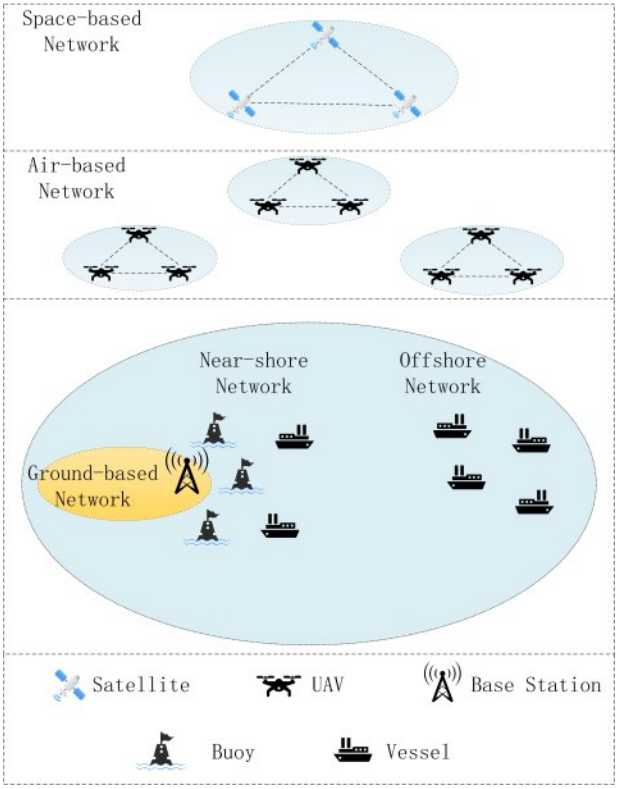

图1　SAGSIN架构

2.1　通信模型

在该场景中，由于节点间距离较远，海上通信衰减较大，因此UAV间和星间干扰可以忽略[18]。

(1) *海上通信*

与陆地通信不同，海面上的散射波比障碍物的影响更大。海上通信的路径损耗可以通过弯曲的地球双射线反射模型[19]来计算[14]，如式(1)。

$$l(d;H_{\mathrm{t}},H_{\mathrm{r}})=\left(\frac{\lambda}{4\pi d}\right)^2\left(2\sin\left(\frac{2\pi H_{\mathrm{t}}H_{\mathrm{r}}}{\lambda d}\right)\right)^2 \tag{1}$$

其中，λ为波长，H_{t}为发射机天线高度，H_{r}为接收机天线高度，d为发射端和接收端直线距离。

(2) *船舶-BS(V2B, Vehicle-to-BS)信道*

根据香农公式可以得到V2B信道的传输速率和传输时间，如式(2)和式(3)。

$$C_{\mathrm{V2B}}=W_{\mathrm{V2B}}\log_2\left(1+\frac{P_{\mathrm{t}}hg_{\mathrm{v}}g_{\mathrm{B}}l\left(d_{\mathrm{V2B}};H_{\mathrm{v}},H_{\mathrm{B}}\right)}{W_{\mathrm{V2B}}\sigma^2}\right) \tag{2}$$

$$t_{\mathrm{tran}}^{\mathrm{V2B}}=D/C_{\mathrm{V2B}} \tag{3}$$

其中，P_{t}为船舶的发射功率，g_{v}和g_{B}分别为船舶和BS的天线增益，d_{V2B}为船舶与BS的直线距离，H_{v}和H_{B}为船舶和BS的天线高度，h为假设服从参数为1的伽马分布的小尺度衰落，σ^2为噪声功率谱密度，W_{V2B}为V2B信道带宽，D为任务大小。

(3) 船舶-UAV(Vehicle-to-UAV，V2U)信道

根据香农公式，V2U信道的传输速率和传输时间表示为式(4)和式(5)。

$$C_{\mathrm{V2U}} = W_{\mathrm{V2U}} \log_2\left(1 + \frac{P_{\mathrm{t}} h g_{\mathrm{v}} g_{\mathrm{U}} l\left(d_{\mathrm{V2U}}; H_{\mathrm{v}}, H\right)}{W_{\mathrm{V2U}} \sigma^2}\right) \tag{4}$$

$$t_{tran}^{V2U} = D / C_{V2U} \tag{5}$$

其中，g_{U}为UAV的天线增益，d_{V2U}为船舶与UAV的直线距离。W_{V2U}为V2U信道带宽。

(4) BS-浮标(B2b, BS-to-buoy)信道

根据香农公式，B2b信道的传输速率和传输时间表示为式(6)和式(7)。

$$C_{B2b} = W_{\mathrm{B2b}} \log_2\left(1 + \frac{P_{\mathrm{B}} h g_{\mathrm{B}} g_{\mathrm{b}} l\left(d_{\mathrm{B2b}}; H_{\mathrm{B}}, H_{\mathrm{b}}\right)}{W_{\mathrm{B2b}} \sigma^2}\right) \tag{6}$$

$$t_{\mathrm{tran}}^{\mathrm{B2b}} = D / C_{\mathrm{B2b}} \tag{7}$$

其中，P_{B}为BS的发射功率，g_{b}为浮标的天线增益，d_{B2b}为BS与浮标的直线距离，H_{b}为浮标天线高度，W_{B2b}为B2b信道带宽。

(5) 船舶-卫星(V2S, Vehicle-to-Satellite)信道

由于ku波段一直用于卫星通信，它不会干扰其他无线通信系统。但它会受到雨水衰减的极大影响。此外，在执行海上任务的过程中，卫星的飞行距离与其高度相比微不足道，因此可以忽略不计[14]。因此，V2S信道的传输速率和传输时间可表示为式(8)和式(9)。

$$C_{\mathrm{V2S}} = \Lambda W_{\mathrm{V2S}} \log_2\left(1 + \frac{P_{\mathrm{t}} h g_{\mathrm{V}} g_{\mathrm{s}}}{W_{\mathrm{V2S}} \sigma^2}\right) \tag{8}$$

$$t_{\mathrm{tran}}^{\mathrm{V2S}} = D / C_{\mathrm{V2S}} \tag{9}$$

其中，Λ为平均雨衰减率，W_{V2S}为V2S信道带宽，g_{s}为卫星天线增益。

2.2 时延模型和能耗模型

对于本地计算，计算时间为$t_{\mathrm{com}}^{\mathrm{local}} = DF/f_{\mathrm{v}}$，$F$为计算1位数据所需的CPU周期数，$f_{\mathrm{v}}$为船舶CPU周期频率。计算产生的能耗为$e_{\mathrm{com}}^{\mathrm{local}} = \kappa\left(f_{\mathrm{v}}\right)^3 t_{\mathrm{com}}^{\mathrm{local}}$，计算因子$\kappa$由芯片架构决定[20]。

UAV计算时间为$t_{\mathrm{com}}^{\mathrm{UAV}} = DF/f_{\mathrm{u}}$，对应计算产生的能耗表示为$e_{\mathrm{com}}^{\mathrm{UAV}} = \kappa\left(f_{\mathrm{u}}\right)^3 t_{\mathrm{com}}^{\mathrm{UAV}}$，$f_{\mathrm{u}}$为UAV的CPU周期频率。与上述类似，卫星、浮标计算时间与能耗分别为$t_{\mathrm{com}}^{\mathrm{sat}} = DF/f_{\mathrm{s}}$，$e_{\mathrm{com}}^{\mathrm{sat}} = \kappa\left(f_{\mathrm{s}}\right)^3 t_{\mathrm{com}}^{\mathrm{sat}}$，$t_{\mathrm{com}}^{\mathrm{buoy}} = DF/f_{\mathrm{b}}$，$e_{\mathrm{com}}^{\mathrm{buoy}} = \kappa\left(f_{\mathrm{b}}\right)^3 t_{\mathrm{com}}^{\mathrm{buoy}}$，$f_s$和$f_b$分别为卫星和浮标的CPU周期频率。

由于计算结果的尺寸较小，可忽略返回时间和能量消耗。因此完成一个任务的总时延表示为式(10)。

$$\begin{aligned} t_{\mathrm{total}} &= \alpha t_{\mathrm{com}}^{\mathrm{total}} \\ &+\beta\left(t_{\mathrm{tran}}^{\mathrm{V2U}} + t_{\mathrm{com}}^{\mathrm{UAV}}\right) \\ &+\gamma\left(t_{\mathrm{tran}}^{\mathrm{V2S}} + t_{\mathrm{com}}^{\mathrm{sat}}\right) \\ &+\theta\left(t_{\mathrm{tran}}^{\mathrm{V2B}} + t_{\mathrm{tran}}^{\mathrm{V2B}} + t_{\mathrm{tran}}^{\mathrm{buoy}}\right) \\ &+t_{\mathrm{wait}} \\ &\alpha + \beta + \gamma + \theta = 1 \\ &\alpha,\beta,\gamma,\theta \in \{0,1\} \end{aligned} \tag{10}$$

其中$\alpha,\beta,\gamma,\theta$均为决策变量，$\alpha = 1$表示任务在本地处理；$\beta = 1$表示任务被卸载到UAV上处理；$\gamma = 1$表示任务被卸载到卫星上处理；$\theta = 1$表示通过BS将任务卸载到浮标上处理。$t_{\mathrm{wait}}$表示为任务进入计算队列后仍需等待的时间。卸载决策向量可表示为$D = \{\alpha,\beta,\gamma,\theta\}$。当$t_{\mathrm{total}} \leqslant t_{\max}$时，代表当前任务卸载成功，$t_{\max}$为任务最大要求时延。

完成一个任务的总能耗e_{total}表示为式(11)。

$$\begin{aligned} e_{\mathrm{total}} &= \alpha e_{\mathrm{com}}^{\mathrm{local}} \\ &+\beta\left(e_{\mathrm{com}}^{\mathrm{UAV}} + P_{\mathrm{t}} t_{\mathrm{tran}}^{\mathrm{V2U}}\right) \\ &+\gamma\left(e_{\mathrm{com}}^{\mathrm{sat}} + P_t t_{\mathrm{tran}}^{\mathrm{V2S}}\right) \\ &+\theta\left(e_{\mathrm{com}}^{\mathrm{buoy}} + P_t t_{\mathrm{tran}}^{\mathrm{V2B}} + P_{\mathrm{B}} t_{\mathrm{tran}}^{\mathrm{B2b}}\right) \end{aligned} \tag{11}$$

任务卸载成功率sr可以表示为式(12)。

$$\mathrm{sr} = \frac{task_{\mathrm{s}}}{task_{\mathrm{total}}} \tag{12}$$

其中，$task_{\mathrm{s}}$为卸载成功任务数，$task_{\mathrm{total}}$为总任务数。

2.3 问题模型

本文设计的2个优化目标分别是最大化任务卸载成功率、最小化能耗，如式(13)所示。

$$P_1: \max_D \mathrm{sr} \tag{13a}$$

$$P_2: \min_D e_{\mathrm{total}} \tag{13b}$$

$$s.t.\ \alpha,\beta,\gamma,\theta \in [0,1] \tag{13c}$$

$$\alpha + \beta + \gamma + \theta = 1 \tag{13d}$$

$$t_{\mathrm{com}}^{\mathrm{local}} \leqslant t_{\max} \tag{13e}$$

$$t_{\mathrm{tran}}^{\mathrm{V2U}} + t_{\mathrm{com}}^{\mathrm{UAV}} \leqslant t_{\max} \tag{13f}$$

$$t_{\mathrm{tran}}^{\mathrm{V2S}} + t_{\mathrm{com}}^{\mathrm{sat}} \leqslant t_{\max} \tag{13g}$$

$$t_{\mathrm{tran}}^{\mathrm{V2B}} + t_{\mathrm{tran}}^{\mathrm{B2b}} + t_{\mathrm{con}}^{\mathrm{buoy}} \leqslant t_{\max} \tag{13h}$$

其中，(13c)和(13d)是与任务卸载决策相关的约束，(13e)~(13h)是时延约束。

3 基于MADDPG的任务卸载方案

本文将每个船舶尝试不同卸载目标的问题建模为一个多智能体强化学习问题，采用马尔可夫决策过程(Markov Decision Process，MDP)建模。在MDP中，智能体在离散的时间步中与环境交互，核心要素包括：状态、动作、奖励函数、状态转移函数、折扣因子。其核心特性是马尔可夫性质，即下一状态和奖励只依赖于当前状态和采取的行动，与历史状态无关。每个船舶作为一个智能体，通过与环境交互学习最优卸载策略。多个智能体共同探索环境，根据环境状态的变化选定任务的卸载目标。智能体共用一个奖励函数，还会因其他智能体任务完成的程度获取额外奖励，借此将智能体间的博弈转化为合作。多智能体强化学习算法涵盖两个阶段：集中式学习和分布式执行阶段。在集中式学习阶段，每个智能体都可以获得奖励，然后通过集中式学习训练Critic和Actor网络。到了分布式执行阶段，各智能体从各自的环境中获得相应的状态，接着用自己训练的Actor网络选择执行的动作。

3.1 状态空间

在时间步t，有M个智能体，智能体k获得的环境状态为S_t^k，智能体根据策略输出动作A_t^k。在所有智能体执行动作后，获得奖励R_t，并根据动作的选择进入对应的下一个状态S_{t+1}^k。对于智能体k而言，只能观察到自己所处环境状态和输出的动作，其他智能体的动作都是未知的。智能体k观测的环境信息包括自身位置p_t^k、自身任务信息$task_t^k$、自身处理任务队列信息q_t^k、卫星处理任务队列信息q_t^s、UAV处理任务队列信息q_t^{UAV}。如果智能体处于近海场景中，那么还可以观测到浮标处理任务队列信息q_t^{buoy}。相反，如果智能体处于远海领域中，则无法观测到浮标处理任务队列信息，而为保证状态维度相同，因此将q_t^{buoy}替换为空信息。因此智能体k所观察到的环境状态为$S_t^k=\left\{p_t^k,task_t^k,q_t^k,q_t^s,q_t^{UAV},q_t^{buoy}\right\}$。

3.2 动作空间

智能体的任务处理决策根据场景的不同会有略微差别，大体上分为两类，一类是本地处理，另一类是卸载到UAV、卫星处理。在近海场景中有浮标部署，因此在此场景中的智能体还可以将计算任务卸载到浮标上处理。因此近海场景中将卸载目标离散为$a=0$、1、2、3，而远海场景中将卸载目标离散为$a=0$、1、2。因此智能体k的动作决策表示为$a_t^k=\{a\}$。

3.3 奖励函数

奖励函数的设计至关重要，它直接影响智能体在高维复杂场景中解决难题的能力。符合目标的奖励能有效提升系统性能。当任务卸载成功后智能体会获得一个基础奖励值r_0，在此基础上减去权重λ_1乘以卸载能耗e和λ_2乘以卸载目标队列长度q_t，以此来鼓励智能体选择当前能耗更小和时延更短的卸载目标。任务卸载失败情况分为两种，卸载时延超出了最大容忍时延或选择了超过通信范围R的卸载目标，即$d>R$，对此分别给予智能体奖励值r_1和r_2。同时，引用一个权重w乘以其他智能体的奖励，来鼓励智能体在环境中进行合作。由此，在时间步t智能体k的奖励表示为式(14)。

$$R_t^k=\begin{cases} w*\sum_{n=0}^{M,M\neq k}R_t^n+r_0-\lambda_1*e_t^k-\lambda_2*q_t, t_{total}^k\leqslant t_{\max}, d\leqslant R \\ w*\sum_{n=0}^{M,M\neq k}R_t^n+r_1, t_{\text{total}}^k>t_{\max}, d\leqslant R \\ w*\sum_{n=0}^{M,M\neq k}R_t^n+r_2, d>R \end{cases} \tag{14}$$

3.4 MADRL算法

(1) 深度确定性策略梯度(DDPG, Deep Deterministic Policy Gradient)算法

DQN算法用神经网络替代Q表，通过其非线性映射能力处理高维状态，解决了状态空间维度爆炸问题。策略梯度(PG, Policy Gradient)算法则直接参数化并优化策略，突破了值函数限制，能处理连续动作空间，并通过采样近似梯度提高效率。DDPG算法结合了DQN和PG算法的优点，构建了双网络协同架构：Actor和Critic网络。Actor网络通过与环境交互执行动作，并在Critic网络价值函数指导下通过策略梯度学习更好的策略。Critic网络利用Actor网络与环境交互收集的数据学习价值函数，来判断当前状态下哪些动作更优，从而辅助Actor进行策略更新。DDPG算法状态价值函数如式(15)所示。

$$Q_k(S_k,A_k)=E\left[R_k+\gamma Q_k(S_{k'},A_{k'})\right] \tag{15}$$

其中，S_k为智能体k的状态输入，A_k为智能体k的动作输入，R_k为智能体k获得的奖励。

DDPG算法涉及4个网络，Actor和Critic网络

各有一个评估网络，Actor和Critic评估网络参数分别为θ_k^A和θ_k^C。通过从经验池中抽取一定数量样本经验并输入到智能体的方式，评估网络参数实现了实时更新。训练中，Actor和Critic网络根据输入的经验更新网络参数。Critic网络通过调整评估网络参数来减少损失。然而，这可能导致神经网络训练不稳定。Critic网络的目标是最小化预测值和目标值间的损失。而目标值是基于当前的Actor和Critic网络计算得到的。当评估网络参数不断更新时，目标值也会随之不断变化。Actor网络根据Critic网络的反馈更新自己的参数，又会进一步影响Critic网络的目标值。这种相互依赖和动态变化使目标值快速变化，这会使评估网络在训练中不断追逐一个变化的目标，评估网络可能会在不同的参数之间来回跳动，无法找到一个合适的参数组合来最小化损失函数，即难以收敛到一个稳定的解，从而导致训练的不稳定。为了解决这一问题，Actor和Critic网络增加了目标网络。Actor和Critic目标网络的参数为$\theta_k^{A'}$和$\theta_k^{C'}$。目标网络的参数采用软更新方式，使目标网络缓慢更新并逐渐接近评估网络。软更新方式如式(16)和式(17)所示。

$$\theta_k^{A'} = \tau\theta_k^A + (1 - \tau)\theta_k^{A'} \tag{16}$$

$$\theta_k^{C'} = \tau\theta_k^C + (1 - \tau)\theta_k^{C'} \tag{17}$$

其中τ表示大小在0到1之间的参数。

(2) Gumbel-Softmax

DDPG算法要求智能体的动作对其策略参数可导，这适用于连续动作空间，但在任务卸载的离散动作空间中不适用。为使离散分布的采样可导，可采用Gumbel-Softmax。通过引入重参数因子g_i，该因子采样自Gumbel(0,1)的噪声，表示为式(18)。

$$g_i = -\log(-\log u), u \sim Uniform(0,1) \tag{18}$$

Gumbel-Softmax采样如式(19)所示。

$$y_i = \frac{\exp\left(\left(\log a_i + g_i\right)/\mu\right)}{\sum_{j=1}^{k}\exp\left(\left(\log a_j + g_i\right)/\mu\right)}, i = 1,\ldots,k \tag{19}$$

其中$a_i \in \xi(a_1,\cdots,a_k)$，$\xi$为某一离散分布。

通过$z = \arg\max_i y_i$计算离散值，该离散值就近似等于离散采样$z \sim \xi$。

(3) MADDPG算法

MADDPG算法为每个智能体实现一个DDPG算法。所有智能体共享一个中心化Critic网络，该网络在训练时为每个智能体的Actor网络提供指导，而各个智能体的Actor网络在执行时独立行动。从实现了去中心化执行。MADDPG框架如图2所示。

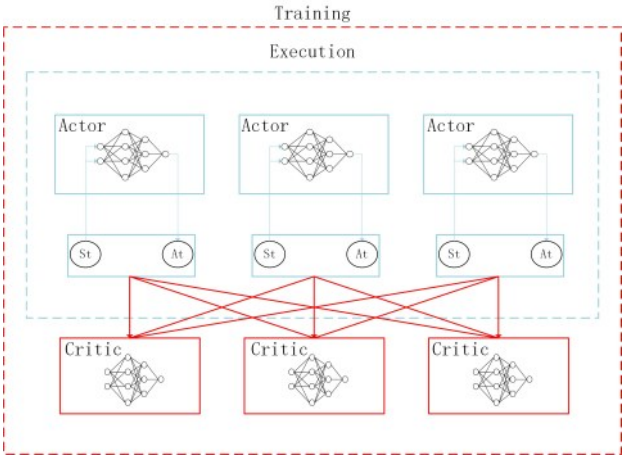

图2　MADDPG框架

MADDPG算法在集中式离线训练阶段，除智能体k观察到的状态S_t^k和智能体的动作A_t^k外，还需引入其他智能体的状态信息和动作信息，并将这些信息合并后存入当前智能体的经验池，用于集中训练Critic网络。此时，用于集中式训练的联合动作表示为A_t，联合状态表示为S_t。借助这些额外信息，每个智能体可分别学习其函数。

该算法的时间复杂度为$O(E*T*M)$，其中E是训练代数，T是每代的步长。这主要由三层嵌套循环决定：最外层E代，中层T次时间步循环，内层对M个智能体依次进行相关操作。

3.5　基于DQN任务卸载方案

基站选择浮标卸载的环境采用DQN算法完成浮标选择。其最终目标与式(13a)、(13b)相同。

3.5.1　状态空间

在时间步t，智能体为基站，智能体状态表示为s_t，智能体根据策略输出动作a_t。智能体执行动作后，从环境中获得奖励r_t。智能体观测的环境信息包括所有浮标与智能体的距离集合d_b、所有浮标队列长度集合q_t。因此智能体k所观察到的状态为$s_t = \{d_b, q_t\}$。

3.5.2　动作空间

智能体的任务被卸载到浮标计算。假设有b个浮标，将卸载目标离散化为$a = 0,1,\cdots,b$。因此智能体动作决策表示为$a_t = \{a\}$。

算法1 基于MADDPG的任务卸载方案

初始化每个智能体的Actor和Critic的评估和目标网络参数
初始化每个智能体的经验池大小B_k
for i = 0 to E － 1 do
  重置环境
  for t = 0 to T － 1 do
    for k = 0 to M － 1 do
      观测智能体k环境获得状态S_t^k
      智能体k选择执行的动作A_t^k
      if A_t^k为卸载到浮标
        使用DQN算法选择浮标卸载
      end if
      智能体k执行动作后获得下一时刻的状态S_{t+1}^k
    end for
    所有智能体执行完动作和获得奖励R_t
    for k = 0 to M － 1 do
      if 经验池内存储经验数量少于B_k do
        存储$\{S_t,A_t^1,\cdots,A_t^k,R_t,S_{t+1}\}$到智能体$k$的经验池中
      else
        用$\{S_t,A_t^1,\cdots,A_t^k,R_t,S_{t+1}\}$替换最早存储的经验
        随机选择大小为B_b的经验样本训练网络
        通过最小化损失函数更新Critic评估网络
        通过最大化策略目标函数来更新Actor评估网络
        根据式(16)更新Actor目标网络
        根据式(17)更新Critic目标网络
      end if
    end for
  end for
end for

3.5.3 奖励函数

为鼓励智能体探索最优卸载策略，通过将奖励值取负并权衡卸载时延与能耗，使智能体为追求奖励值最大。因此奖励函数设计如式(20)所示。

$$r_t = -\left(t_{\text{tran}}^{buoy} + t_{\text{com}}^{buoy} + \rho_1 * e^{buoy} + \rho_2 * q^{buoy}\right) \tag{20}$$

其中ρ_1,ρ_2为0到1的权重。

3.5.4 DQN算法

DQN算法在Q-learning算法基础上引入神经网络，借助函数拟合估计Q值，解决了难以应对大规模状态和动作数量的问题。DQN算法同样采用评估和目标网络，但目标网络参数更新是每隔固定训练步数与评估网络同步一次，而非采用软更新方式。

该算法的时间复杂度为$O(E*T)$，由两层嵌套循环决定：外层循环执行E代，内层循环执行T个时间步。

算法2 基于DQN任务卸载方案

初始化智能体网络参数
初始化智能体的经验池大小B
for i = 0 to E － 1 do
  重置环境
  for t = 0 to T － 1 do
    观测智能体环境获得状态s_t
    智能体选择动作a_t并获得奖励r_t
    智能体执行动作后获得下一时刻的状态s_{t+1}
    if 经验池内存储经验数量少于B do
      存储$\{s_t,a_t,r_t,s_{t+1}\}$到经验池中
    else
      用$\{s_t,a_t,r_t,s_{t+1}\}$替换最早存储的经验
      随机选择大小为B_b的经验样本训练网络
      if 经过N个训练步
        更新目标网络
      end if
    end if
  end for
end for

4 实验结果及分析

4.1 实验环境

SAGSIN场景的主要参数如表1所示。每个智能体的网络均由3个全连接的隐藏层组成。激活函数采用修正线性单元，优化器采用适应性矩估计来迭代训练并更新神经网络权重。由于所有任务均为计算密集型，因此选择较小规模的任务，任务大小范围为0.5 M至1.0 M，并将其平均分为5个区间。

4.2 基线方案

(1) MADQN[22-24]方案：本算法单个智能体遵循DQN[25-27]算法，DQN算法仅有Q网络。

(2) DDPG[28-31]方案：算法中只有一个智能体，智能体只能观测单个船舶的状态和奖励

(3)随机方案：在每个时间步，船舶随机选择一个卸载目标。

4.3 实验结果

4.3.1 算法收敛性分析

训练代数增长对奖励的影响如图3所示。每代中有100个训练步，算法在前500代不参与训练，仅收集经验，在第501代开始训练。

由图3可看出：第1000代时，算法开始趋于收敛；由于任务大小的随机性以及小尺度衰落的随机性，导致算法收敛时存在数值波动。

表 1　实验参数设置

参数	取值
UAV 高度 H	500m
天线高度 H_V,H_B,H_b	5m,50m,1m
安全距离 r	50m
通信距离 R	800m
传输功率 P_t,P_B,P_H	30dBm,40dBm,40dBm
信道带宽 W_{V2B},W_{V2U},W_{V2S},W_{B2b}	2.5MHz,2MHz,5MHz,2.5MHz
天线增益 g_v,g_B,g_U,g_b,g_s	41dBi,41dBi,30dBi,30dBi,41dBi
平均雨衰减率 Λ	0.9[21]
计算因子 κ	10^{-27}
CPU 周期频率 f_u,f_h,f_s,f_b	1GHz,1.5GHz,3GHz,2GHz
噪声功率谱密度 σ^2	-160dBm/Hz
计算 1 位数据所需的 CPU 周期数 F	$3.3*10^3$cycle/bit
UAV 密度 λ_U	$2*10^{-5}/m^2$
权重因子 λ_1,λ_2,w,ρ_1,ρ_2	0.2,0.3,0.1,0.3,0.5
奖励值 r_0,r_1,r_2	6,-2,-4
经验池大小 B	50000
样本数量 B_b	32
训练代数 E	1500
训练步数 T	100
软更新参数 τ	0.98

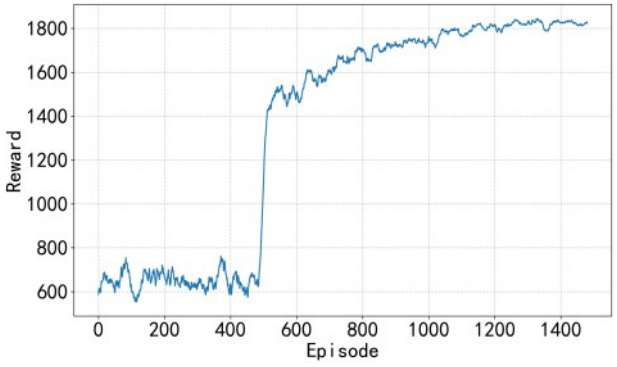

图 3　MADDPG 算法收敛性

DQN 算法训练代数增长对奖励的影响如图 4 所示。算法从第 1 代开始训练，共训练 1500 代。

由图 4 可看出：算法从第 500 代趋于收敛；同样由于任务大小的随机性以及小尺度衰落的随机性，导致算法收敛时存在数值波动。

4.3.2　近海远海场景资源利用分析

在本文场景中，充分利用近海场景和远海场景的资源是实现所提目标的关键。近海场景拥有更多的计算资源，而远海场景资源相对匮乏，更考验智能体的合作与资源协调。

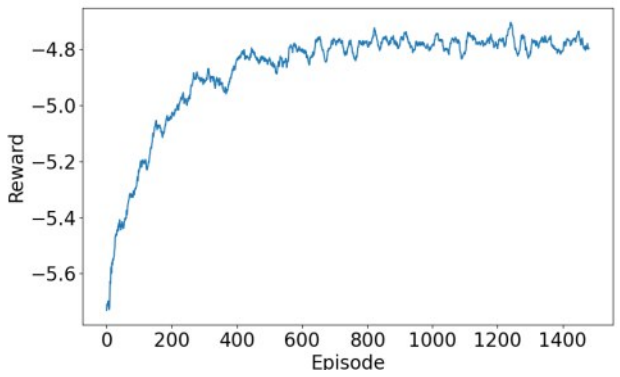

图 4　DQN 算法收敛性

近远海场景的卸载决策占比如图 5、图 6 所示。

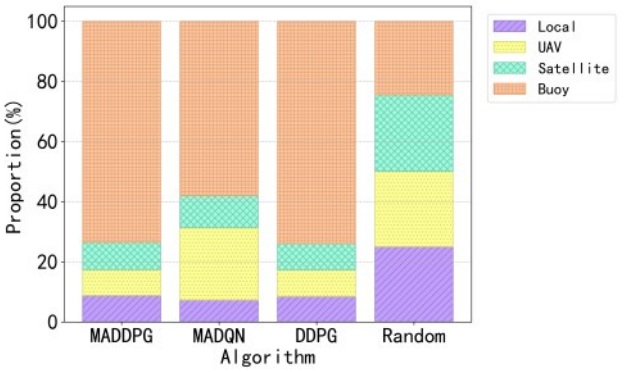

图 5　近海场景不同算法卸载决策占比

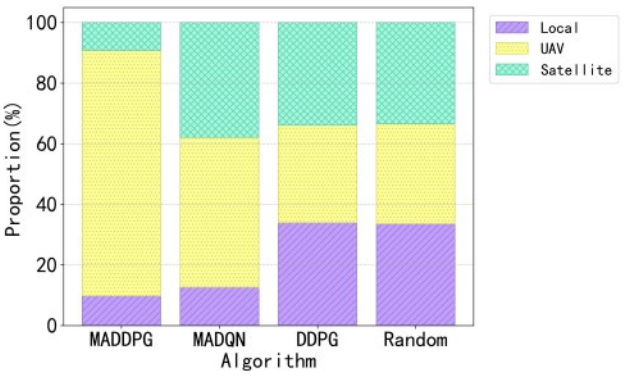

图 6　远海场景不同算法卸载决策占比

由图 5、图 6 可看出：本文所提卸载方案在近海场景中优先利用浮标资源，大幅降低卫星依赖；在远海场景中动态重构资源优先级，将无人机作为核心替代节点，并严格限制高能耗卫星使用，使得全场景覆盖的卫星资源在充分利用的同时不过度使用，从而在两类场景中均实现最优卸载决策。

4.3.3　任务卸载成功率分析

所提算法卸载方案与三种基线卸载方案任务卸

载成功率训练曲线如图5所示。DDPG算法同样在前500代不参与训练，仅收集经验，在第501代开始训练。而MADQN算法则从第1代开始训练。

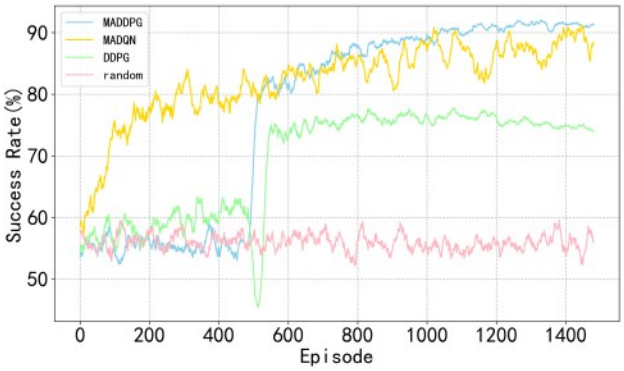

图7　训练代数变化对卸载成功率的影响

由图7可看出：MADDPG算法的卸载成功率从第1000代开始趋于平稳。而MADQN算法波动较大，稳定性不如MADDPG。DDPG算法在训练后期虽然比MADQN更稳定，但其卸载成功率相对更低。随机方案的卸载成功率一直保持较低状态。通过对不同策略方案训练1500代的卸载成功率增长情况进行对比，可以看出所提算法具有更强的鲁棒性，从而证明了其优越性。

不同任务大小区间的任务卸载成功率比较如图8所示。

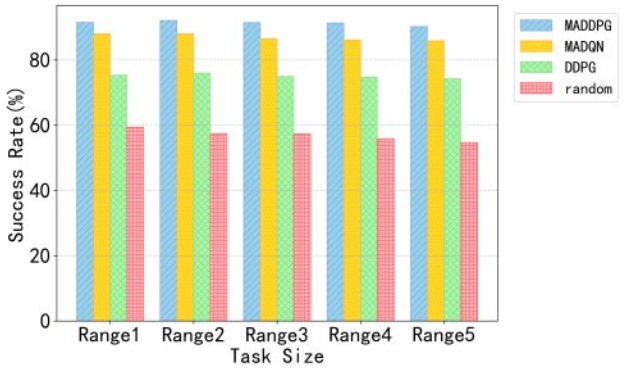

图8　任务大小变化对卸载成功率的影响

由图8可看出：任务数据量的动态变化对各方案成功率产生差异化影响。本文提出的卸载方案在全任务规模区间内均展现出最高卸载成功率。MADDPG通过多智能体协作机制在任务数据量波动时保持了稳定的决策性能，其平均成功率较MADQN方案、DDPG方案和随机方案各提升5.08%、21.71%和60.48%，验证了其在不同任务数据环境下的鲁棒性和泛化能力。

4.3.4　任务卸载时延对比

在本文场景中，追求更高的卸载成功率也代表追求更低的任务卸载时延。任务数据量的不同会直接影响到各个方案的卸载时延。

近海场景、远海场景和全场景的不同任务大小的卸载时延如图9、图10、图11所示。

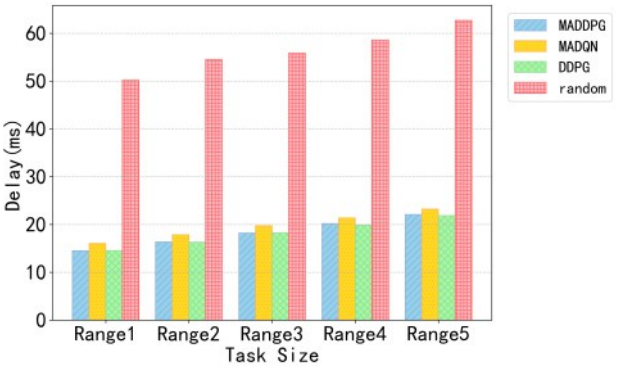

图9　近海场景任务类型变化对时延的影响

由图9可以看出：在计算资源相对充裕的近海场景下，除随机方案外，其余方案均能有效地寻找到具有较低卸载时延的卸载策略。其余方案在各个任务规模点上的卸载时延与随机方案相比均降低了65.28%以上。这充分表明在丰富的计算资源支撑下，合理算法选择对优化卸载时延具有关键作用，而随机方案由于缺乏针对性的优化逻辑，导致其在卸载时延表现上相对逊色，无法有效利用近海场景的计算资源优势。

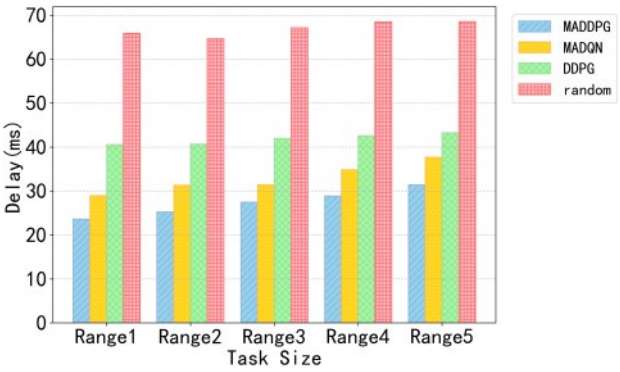

图10　远海场景任务类型变化对时延的影响

由图10可看出：在资源受限较高的远海场景中，各方案的时延表现呈现出显著差异。本文提出的方案在全任务规模区间内均实现了最低的卸载时延，这表明智能体之间通过MADRL协作成功地探索出了更为高效的卸载策略组合。MADDPG算法在计算资源匮乏的约束条件下优化任务分配，其平

均卸载时延较其余方案分别降低16.87%、34.80%、59.27%。这充分验证了MADDPG方案在远海环境下的适应性优势。

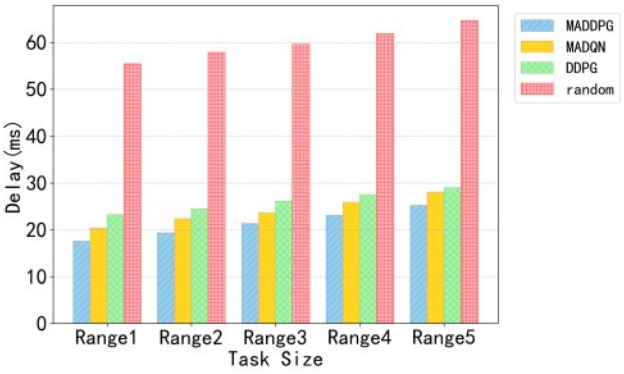

图11　全场景任务类型变化对时延的影响

由图11可看出：所提方案的卸载时延优于其他方案，整体表现最优。具体而言，MADDPG通过多智能体之间的信息共享与联合策略优化，在复杂任务分配场景中实现了任务卸载时延的全局最小化。

4.3.5　任务卸载能耗对比

在本文场景中，在保证卸载成功率的基础上，更低的卸载能耗为第二目标，任务数据量的不同会直接影响到各个卸载方案的卸载能耗。近海场景、远海场景和全场景的不同任务大小的卸载能耗如图12、图13、图14所示。

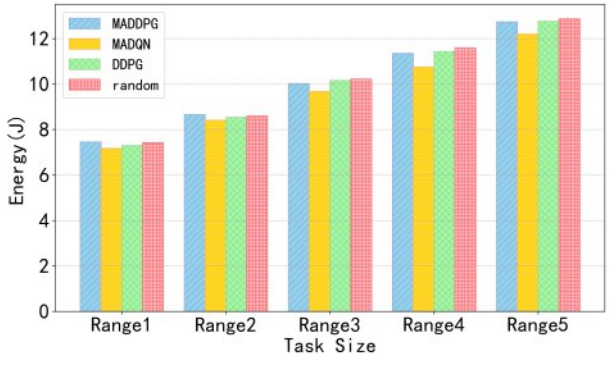

图12　近海场景任务类型变化对能耗的影响

由图12可看出：在近海场景中，各方案的能耗表现相对接近，整体能耗水平差异较小。而MADQN方案卸载能耗相较于本文所提方案平均降低3.90%。这一现象可归因于所提方案在优化目标上的权衡策略：为实现更高的卸载成功率和更低的卸载时延，所提方案在资源分配过程中选择了计算和通信资源投入更高的处理策略。如图9所示，所

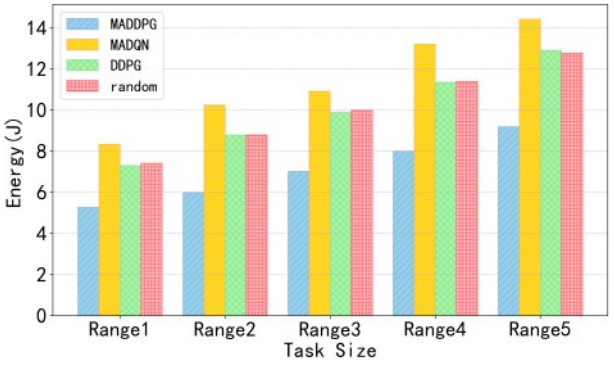

图13　远海场景任务类型变化对能耗的影响

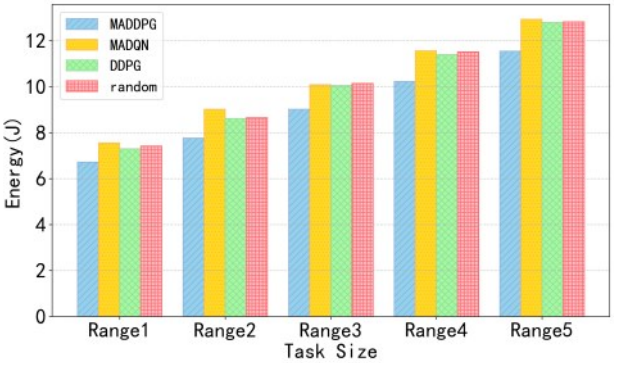

图14　全场景任务类型变化对能耗的影响

提方案的平均卸载时延相较于MADQN方案降低了7.25%，验证了所提方案在近海场景中通过适度增加能耗以换取综合性能提升的有效性，为SAGSIN中的任务调度提供了更具适应性的解决方案。

由图13可看出：在远海场景中，MADDPG算法方案展现出了显著的优势，其卸载能耗在全任务规模区间均为最低。这一现象可归因于多智能体协作机制与远海场景资源约束的协同优化效应。具体而言，远海场景中计算资源的匮乏促使智能体之间必须通过高效协作来实现资源的精准分配。所提方案的平均卸载能耗较其余方案分别降低37.97%、29.42%、29.70%。证明了所提方案在有限资源条件下动态平衡各智能体的策略更新，避免了单点资源过载导致的能耗冗余。

由图14可看出：MADDPG方案的卸载能耗显著优于其他方案，整体表现最优。这一成果归功于MADDPG算法强大的多智能体协作能力。证明了所提方案为SAGSIN中的任务卸载提供了高效、节能的智能决策方案，助力构建绿色、可持续的M-IOT架构。

4.3.6　任务卸载时延与卸载能耗关系

在本文场景中，计算任务卸载时延与能耗消耗

是两个相互冲突的目标，如图13、图14所示。

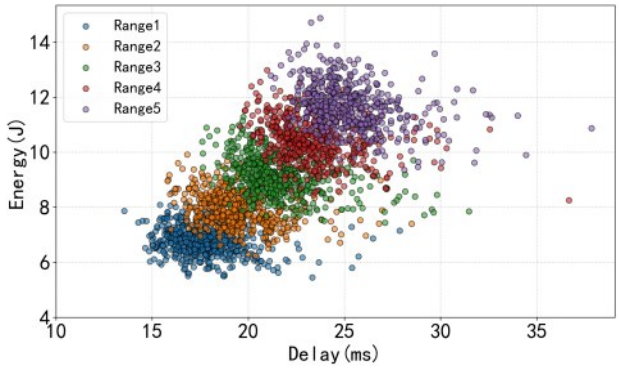

图15 MADDPG算法方案卸载时延与能耗的关系

由图15可看出：随着任务规模的逐步增大，卸载时延与能耗呈现出正相关变化趋势。MADDPG方案的任务数据分布特性表明，不同任务区间内的数据点呈现出明显的聚集效应，且各区间内的时延与能耗波动幅度较小。具体而言，MADDPG算法使智能体在不同任务区间内自适应调整资源分配比例，有效平衡了任务负载与资源消耗，使其能够在复杂任务场景中始终保持高效的任务卸载性能。

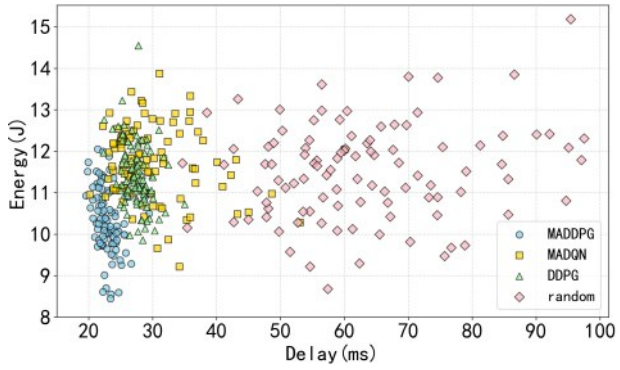

图16 相同任务区间卸载时延与能耗的关系

由图16可看出：相同任务区间内，所提方案相较于其他方案展现出显著的性能优势。MADDPG算法通过高效的MADRL框架，在保证任务卸载质量的前提下，实现了能量消耗的最小化。其能源效率的提升主要归功于智能体间的协作优化机制，使得各智能体能够在有限的资源约束下，动态调整任务分配策略，从而降低整体能耗。与其他方案相比，MADDPG算法在相同时间消耗内，能够找到更优的解，展现出更强的优化能力和更高的资源利用率。

5 结论

近年来，5G的发展以及海上网络资源的不足推动了SAGSIN的发展。本文利用深度强化学习解决近海和远海领域的任务卸载问题，以最大化任务卸载成功率和最小化卸载能耗为目标，采用马尔可夫决策过程进行建模。接着利用MADDPG算法获取最优任务卸载策略。最后，通过一系列仿真实验验证解决方案的有效性。但是算法的时间复杂度与问题规模相关，随着问题规模越来越大算法的时间复杂度也会越来越高，并且所提算法网络结构复杂，因此训练时间等成本开销与其他算法相比更高。

未来研究将关注内容放置与边缘缓存问题，通过提前在边缘服务器下载任务所需的计算资源，以提升卸载任务的计算效率。

参考文献：

[1] 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 J. Commun, 2023, 4: 6 – 78.

[2] 李智勇，王琦，陈一凡，等. 车辆边缘计算环境下任务卸载研究综述[J]. 计算机学报，2021，44(05): 963-982.
LI Z Y, WANG Q, CHEN Y F, et al. A Survey on Task Offloading Research in Vehicular Edge Computing[J]. Chinese Journal of Computers, 2021, 44(05): 963-982

[3] 刘雷，陈晨，冯杰，等. 车载边缘计算中任务卸载和服务缓存的联合智能优化[J]. 通信学报，2021, 42(01): 18-26.
LIU L, CHEN C, FENG J, et al. Joint intelligent optimization of task offloading andservice caching for vehicular edge computing [J]. Journal on Communications, 2021, 42(01): 18-26

[4] ZHANG J, GUO H Z, LIU J J, et al. Task Offloading in Vehicular Edge Computing Networks: A Load-Balancing Solution[J]. IEEE Transactions on Vehicular Technology, 2020, 69(2): 2092-2104.

[5] HEVESLI M, SEID A M, ERBAD A, et al. Multi-Agent DRL for Queue-Aware Task Offloading in Hierarchical MEC-Enabled Air-Ground Networks[J]. IEEE Transactions on Cognitive Communications and Networking, 2025.

[6] BARICK S, SINGHAL C. UAV-Assisted MEC Architecture for Collaborative Task Offloading in Urban IoT Environment[J]. IEEE Transactions on Network and Service Management, 2025, 22 (1): 732-743.

[7] CHEN Z Y, HUANG Z Q, ZHANG J J, et al. Resource Allocation and Collaborative Offloading in Multi-UAV-Assisted IoV With Federated Deep Reinforcement Learning[J]. IEEE Internet of

Things Journal, 2024, 12(5): 4629-4640.

[8] CHAI Z Y, KANG H S, LI Y L, et al. Computation Offloading for Integrated Satellite-Terrestrial Internet of Vehicles in 6G Edge Network: A Cooperative Stackelberg Game[J]. IEEE Transactions on Intelligent Transportation Systems, 2024, 25(8): 10389-10404.

[9] LAN W J, CHEN K Y, CAO J N, et al. Security-Sensitive Task Offloading in Integrated Satellite-Terrestrial Networks[J]. IEEE Transactions on Mobile Computing, 2025, 24(3): 2220-2233.

[10] LI A, ZHOU T, XU T H, et al. LEO Satellite Assisted Edge Computing with Latency and Energy Optimization[J]. IEEE Transactions on Network Science and Engineering, 2025.

[11] 曾锋，张政，陈志刚.基于深度强化学习的计算卸载与资源分配策略[J].通信学报，2023, 44(07):124-135.
ZENG F, ZHANG Z, CHEN Z G. Computation offloading and resource allocation strategybased on deep reinforcement learning[J]. Journal on Communications, 2023, 44(07): 124-135.

[12] LI J L, SHI W S, WU H Q, et al. Cost-Aware Dynamic SFC Mapping and Scheduling in SDN/NFV-Enabled Space–Air–Ground-Integrated Networks for Internet of Vehicles[J]. IEEE Internet of Things Journal, 2022, 9(8): 5824-5838.

[13] XIE W X, CHEN C, JU Y, et al. Deep Reinforcement Learning-Based Computation Computational Offloading for Space–Air–Ground Integrated Vehicle Networks[J]. IEEE Transactions on Intelligent Transportation Systems, 2025, 26(5): 5804-5815.

[14] LIN Z J, YANG J J, CHEN Y Y, et al. Maritime Distributed Computation Offloading in Space-Air-Ground-Sea Integrated Networks [J]. IEEE Communications Letters, 2024, 28(7): 1614-1618.

[15] XU F M, YANG F, ZHAO C L, et al. Deep reinforcement learning based joint edge resource management in maritime network[J]. China Communications, 2020, 17(5): 211-222.

[16] MENG S Q, WU S H, WU H Y, et al. Effectiveness-Oriented SAGSIN: Unveiling a Unified Metric and a Comprehensive Framework[J]. IEEE Network, 2020, 38(6): 39-47.

[17] XU J J, KISHK M A, ALOUINI M S, Space-Air-Ground-Sea Integrated Networks: Modeling and Coverage Analysis[J]. IEEE Transactions on Wireless Communications, 2023, 22(9): 6298-6313.

[18] YU S, GONG X W, SHI Q, et al. EC-SAGINs: Edge-Computing-Enhanced Space Air Ground Integrated Networks for Internet of Vehicles[J]. IEEE Internet of Things Journal, 2022, 9(8): 5742-5754.

[19] WANG J, ZHOU H F, LI Y, et al. Wireless Channel Models for Maritime Communications[J]. IEEE Access, 2018, 6: 68070-68088.

[20] MAO Y Y, YOU C S, ZHANG J, et al. A Survey on Mobile Edge Computing: The Communication Perspective[J]. IEEE Communications Surveys & Tutorials, 2017, 19(4): 2322-2358.

[21] LIN Z J, CHEN X P, CHEN P P. Energy harvesting space-air-sea integrated networks for MEC-enabled maritime Internet of Things [J]. China Communications, 2022, 19(9): 47-57.

[22] LIU Z X, CHEN Y, YUAN Y Z, et al. A Dynamic Power Allocation Scheme Based on Multiagent Deep Q-Network With Environmental Awareness for 5G Dense Networks[J]. IEEE Internet of Things Journal, 2025, 12(4): 4220-4231.

[23] BETALO M L, LENG S P, ABISHU H N, et al. Multi-Agent DRL-Based Energy Harvesting for Freshness of Data in UAV-Assisted Wireless Sensor Networks[J]. IEEE Transactions on Network and Service Management, 2024, 21(6): 6527-6541.

[24] HE J Y, LIU Z Y, ZHANG Y K, et al. Power Allocation Based on Federated Multiagent Deep Reinforcement Learning for NOMA Maritime Networks[J]. IEEE Internet of Things Journal, 2025, 19 (9): 12869-12884.

[25] XIAO H, HU Z G, ZHANG X Y, et al. Federated Deep Reinforcement Learning for Task Offloading in MEC-Enabled Heterogeneous Networks[J]. IEEE Internet of Things Journal, 2025, 12(8): 10238-10252.

[26] ZHAO H, LU G Y, LIU Y, et al. Safe DQN-Based AoI-Minimal Task Offloading for UAV-Aided Edge Computing System[J]. IEEE Internet of Things Journal, 2024, 11(19): 32012-32024.

[27] WANG X, LV J H, SLOWIK A, et al. Augmented Intelligence of Things for Priority-Aware Task Offloading in Vehicular Edge Computing[J]. IEEE Internet of Things Journal, 2024, 11(22): 36002-36013.

[28] MIN J, JIAN W, LIANG Z, et al. DDPG-based intelligent computation offloading and resource allocation for LEO satellite edge computing network[J]. China Communications, 2025, 22(5): 1-15.

[29] CHEN H S, CUI H X, WANG J H,et al. Computation Offloading Optimization for UAV-Based Cloud-Edge Collaborative Task Scheduling Strategy[J]. IEEE Transactions on Cognitive Communications and Networking, 2025.

[30] THAT V, CHHEA K H and LEE J R. Optimizing Energy Consumption and Latency in IoT Through Edge Computing in Air–Ground Integrated Network With Deep Reinforcement Learning [J]. IEEE Open Journal of Vehicular Technology, 2025, 6: 412-425.

[31] DU J B, WANG J X,SUN A J, et al. Joint Optimization in Blockchain- and MEC-Enabled Space–Air–Ground Integrated Networks[J]. IEEE Internet of Things Journal, 2024, 11(19): 31862-31877.