

## 海洋观监测传感器网络多接入边缘计算卸载方法

苏新<sup>1</sup>, 王子怡<sup>1</sup>, 王宇鹏<sup>2</sup>, 周思源<sup>3</sup>

(1. 河海大学物联网工程学院, 江苏 常州 213002; 2. 沈阳航空航天大学电子信息工程学院, 辽宁 沈阳 110136;  
3. 河海大学计算机与信息学院, 江苏 南京 211106)

**摘要:** 多接入边缘计算 (MAC, multi-access edge computing) 可有效保障海洋观监测传感器网络 (简称传感网) 的低时延、高可靠数据传输及其各类相关海事应用。在近海场景下, 结合边缘计算资源分布建立多用户单跳单播 (MSU, multi-user single-hop unicast) 与多用户多跳单播 (MMU, multi-user multi-hop unicast) 两种卸载模型。利用混合整数非线性规划分离优化目标, 有效地分配传输功率, 并通过改进传统人工鱼群算法 (AFSA, artificial fish swarm algorithm) 制定卸载决策。结果表明, 相比传统方案, 所提优化算法可降低网络时延近 19%。在远海场景下, 建立远海 MSU 卸载模型, 结合网络连通概率提出合理的信道分配算法。结果表明, 所提算法在网络连通时间充足时, 可增加允许配子信道数量, 降低网络时延; 在网络连通时间有限时, 可控制卸载的海洋用户设备数量, 保障网络传输时延。

**关键词:** 海洋观监测传感器网络; 多接入边缘计算; 人工鱼群算法; 信道分配

**中图分类号:** TN929.52

**文献标识码:** A

**doi:** 10.11959/j.issn.2096-3750.2021.00205

## Multi-access edge computing offloading in maritime monitoring sensor networks

SU Xin<sup>1</sup>, WANG Ziyi<sup>1</sup>, WANG Yupeng<sup>2</sup>, ZHOU Siyuan<sup>3</sup>

1. College of Internet of Things Engineering, Hohai University, Changzhou 213002, China  
2. College of Electronic and Information Engineering, Shenyang Aerospace University, Shenyang 110136, China  
3. College of Computer and Information, Hohai University, Nanjing 211106, China

**Abstract:** Multi-access edge computing can effectively guarantee the low-latency, high-reliability data transmission of ocean monitoring sensor networks and various related maritime applications. In the offshore scenario, two offloading models of multi-user single-hop unicast and multi-user multi-hop unicast were established in combination with the distribution of edge computing resources. The mixed integer nonlinear programming was used to separate optimization targets and effectively allocate transmission power. The unloading decisions were made by improving the traditional artificial fish swarms algorithm. The results show that the proposed optimization algorithm can reduce the network delay by nearly 19% compared with the traditional scheme. In the far-sea scenario, a multi-user single-hop unicast offloading model was established, and a reasonable channel allocation algorithm was proposed based on the network connection probability. The results show that when the network connection time is sufficient, the number of allowable sub-channels can be increased to reduce the network delay. When the network connection time is limited, the number of unloaded marine user equipment can be controlled to ensure the network transmission delay.

**Key words:** maritime monitoring sensor network, multi-access edge computing, artificial fish swarm algorithm, channel allocation

收稿日期: 2020-12-20; 修回日期: 2021-01-26

通信作者: 苏新, leosu8622@163.com

基金项目: 国家自然科学基金资助项目 (No.61801166); 中央高校基本科研业务费资助项目 (No.B210202091); 中国民用航空局 2019 年度民航安全监管能力建设项目

**Foundation Items:** The National Natural Science Foundation of China (No.61801166), The Fundamental Research Funds for the Central Universities (No.B210202091), The Safety Enhancement Construction Research Fund of Civil Aviation Administration of China

## 1 引言

海洋观监测传感器网络（简称传感网）作为未来海洋信息智慧网络的重要组成部分，可提供多种观监测应用，是汇聚海洋空间、环境、资源等各类数据的重要平台。全天候、全自动的海洋观测和目标态势感知、海洋信息传输以及海上综合业务服务的开展，提出了实时定位、紧急救援以及面向军事领域的低时延信息服务需求。这类需求对海洋观监测传感网支持低时延、高可靠的观监测应用提出了极高要求。面向新一代移动网络的多接入边缘计算（MAC, multi-access edge computing）技术可有效支撑以上各类海事业务和应用需求<sup>[1-3]</sup>，将云端数据中心“降级”至网络边缘，使其能够在靠近用户的位置提供计算、存储和通信的能力<sup>[4]</sup>。然而，基于 MAC 的全新海洋观监测传感网络体系依然面临若干科学问题。

复杂海事观监测应用会导致网络局部区域数据的超负荷处理，部分区域网络开销急剧增长，如何实现海洋观监测传感网资源的灵活适配是待解决的首要问题。

相比传统陆地蜂窝网及车载网，海洋观监测传感网环境因素复杂且节点差异化明显。特别地，近海场景下海洋观监测传感网络节点密度较大，海洋边缘计算能力分布不均，资源调度复杂；远海场景下网络节点密度较小，网络连通性易受天气、恶劣海况等因素影响，服务连续性难以保障。如何提出适应不同海事场景的边缘数据卸载模型与算法也是需要解决的问题。

结合与 MAC 卸载技术相关研究以及海洋观监测传感网组网面临的挑战<sup>[1-3]</sup>，以满足低时延、高可靠的海事应用需求为目标，本文面向海洋观监测传感网中的 MAC 卸载技术开展研究，研究成果及主要贡献如下。

本文面向海洋观监测传感网分别制定近海与远海两种不同场景下的多用户卸载模型。针对近海移动节点计算资源充足与有限两种情况，分别建立 MSU 和 MMU 两种卸载模型。设置相应优先级函数，分别解决 MSU 卸载模型下多用户争夺信道资源与 MMU 卸载模型下的一级卸载节点计算资源差异明显问题。同时，针对移动节点数量、计算资源以及网络连通时间都有限的远海场景，建立远海 MSU 卸载模型。为缓解数据卸载阻塞情况，可利用正交频分多址（OFDMA, orthogonal frequency di-

vision multiple access）技术分配物理信道。

在近海场景下，将所提的混合整数非线性规划优化问题分离成两个子问题，利用数学优化有效分配传输功率，并通过改进传统人工鱼群算法有效制定卸载决策。在远海场景下，提出应用于远海 MSU 卸载的低复杂度子信道分配算法，解决远海网络通信时间充足和受限情况下多用户与子信道最佳匹配问题。

## 2 相关工作

MAC 是未来移动通信的核心技术<sup>[5-7]</sup>，结合该技术优势和海洋网络边缘计算特征，本文提出了一套针对不同海洋场景的优化卸载方案。然而，现阶段将 MAC 应用于海洋观监测传感网以提高网络性能的成果相对匮乏，因此，本文重点围绕基于陆地场景的多接入边缘计算卸载方案研究成果进行分析论述，部分相关研究成果为本文的创新提供了重要参考价值。

### 2.1 单用户多接入边缘计算

单用户多接入边缘计算模型可以很好地适应移动终端局限性，通过提升移动终端计算和存储资源利用率带来更好的用户体验。文献[8]将终端数据全部卸载至边缘服务器，提出了基于预先计算的随机离线策略以优化无线电计算资源调度和卸载。多接入边缘计算的应用不仅能够节省能耗，在降低时延方面效果也很显著。文献[9]提出了基于自适应优化的计算卸载方法，并利用 Lyapunov 理论的自适应算法对时延和能耗加权进行了优化。虽然完全卸载方案相较本地处理在时延和能耗方面得到了改善，但上述方案将终端数据作为整体，只能实现完全本地处理或完全卸载，系统的总效率依然受限。

数据拆分卸载方案有两种形式：一种是将数据划分成多个子任务，每个子任务决定是否卸载<sup>[10-11]</sup>；另一种是引入变量将数据分成能否卸载的两部分<sup>[12]</sup>。文献[10]考虑了不可卸载部分数据的本地处理，而将可卸载部分数据平均分成一定数量的子任务，通过低复杂度的次优算法实现本地计算与卸载之间的能耗差最大化，获取各子任务的最优卸载决策。然而，以上研究成果重点关注了能耗敏感型用户需求，且数据的平均拆分缺乏实际意义。于是，为了满足用户多样化需求，文献[11]提出了多任务时延及能耗的加权和优化问题，并采用了交替求解和求

次优解的方法实现资源分配及卸载决策。文献[12]研究了动态电压定标技术在多接入边缘计算中的应用,引入了变量分割待处理数据。同时提出了降低能耗和缩短时延的方案,以满足不同的用户需求。面对多样化的海事应用,单用户模型不适合未来智慧海洋的发展趋势。日趋复杂的数据迁移环境需要研究多用户模型下资源合理分配问题,以进一步提高用户体验。

## 2.2 多用户多接入边缘计算

文献[13]提出了任务缓冲区稳定性约束下的功耗最小化问题,并通过 Lyapunov 算法优化用户在每个时隙的资源分配。文献[14]将时分多址技术应用于多接入边缘计算卸载模型,提出了一种联合通信和计算资源分配优化算法以降低时延。基于文献[14],文献[15]利用 OFDMA 技术提出一种节能计算卸载和资源分配算法用以优化系统开销。文献[16]结合时分多址与 OFDMA 两种技术的优势建立多用户卸载模型,分析了计算资源有限情况下的能耗优化问题,并提出次优算法以高效分配两种模型下的计算资源。文献[13-16]仅考虑了数据上行链路传输时延,而文献[17]综合考虑了数据上/下行链路传输时延,将时延优化问题分层求解,利用二分法优化内层资源分配,并通过贪婪启发式算法优化外层卸载决策。但该方法的卸载模型忽略了边缘节点计算时延(仅考虑了卸载传输时延)和卸载能耗约束,无法满足实际应用需求。基于文献[17],文献[18]考虑了卸载节点计算时延,通过逐次凸逼近算法得到任务卸载分区和时间分配,降低总能耗的同时增加了算法的计算复杂度。

现有多用户边缘计算卸载模型考虑因素相对简单,将二进制卸载决策变量放宽为连续变量,再利用拉格朗日乘数法优化求得的卸载决策难以应用于实际。同时,在利用启发式算法求解优化问题时忽略了 MAC 服务器计算时延及能耗约束,无法满足海洋观监测传感网各类海事应用需求。基于以上分析,本文针对不同海洋场景,将单用户模型下的两种卸载数据划分方式应用于多用户模型,从而建立灵活性更优的部分卸载模型。此外,利用 OFDMA 技术划分信道进一步实现海洋观监测传感网低时延、高可靠性传输。部分海洋节点(通信、计算)资源有限,因此,本文设计的卸载模型考虑了节点计算时延及传输能耗约束,更加切合海洋观监测传感网特性。

## 3 海洋观监测传感网边缘计算卸载

### 3.1 海洋观监测传感网架构

海洋观监测传感网边缘计算卸载模型如图 1 所示,主要由海洋用户设备(OUE, oceanic user equipment)和海洋边缘计算节点(OECN, oceanic edge computing node)两部分组成。

OUE 本地计算能力与信号覆盖范围受限、对能耗敏感,可全天候、全自动开展海事观监测业务,具体包括轻型无人潜航器、海上浮标、小型船舶、无人驾驶船舶等。轻型无人潜航器可用于港口侦察与监视、雷区探测和特定目的地海底调查等。海上浮标可长期、连续地为海洋科学研究、海上石油(气)开发、港口建设收集所需要的海洋水文水质气象资料。小型船舶具备感知与通信能力,可执行海道测量、海洋环境监测、海洋资源开发等相关任务。无人驾驶船舶可代替潜水员进行沉船打捞、深水勘探以及水下电缆铺设等作业。

OECN 内部能源充足、本地计算能力较强且信号覆盖范围较远,可有效开展相关海事观监测业务,具体包括海岸基站(SBS, seacoast base station)、中大型船舶。中大型船舶可作为边缘计算节点,实时处理 OUE 节点外包任务,也可作为中继节点实现任务转发,并能够接入水面自组织网络以扩展海岸基站网络覆盖范围。海岸基站作为空中接口,可实现海洋立体观监测设备终端与陆地云端服务器的连接,也可作为边缘服务器有效支撑高可靠、低时延的相关业务。

### 3.2 近海场景边缘计算卸载

图 1 描述了近海场景下多任务集实时并行执行场景。此时,海洋网络局部区域会产生大量需要实时处理的数据,这些数据可以融合聚类分别卸载至附近的 OECN 进行处理。如果 OUE 邻近海岸基站,可直接卸载任务至相邻基站,称为一级 MSU 卸载。如果 OUE 远离海岸基站,可将各类任务卸载至附近中大型船舶进行实时处理,也可结合自身服务器实时负载情况卸载部分任务至海岸基站进行处理,称为二级 MMU 卸载。

一级 MSU 卸载场景适用于海洋网络边缘计算资源充足的情况,并考虑近海场景下  $K$  个 OUE  $k \in \{1, 2, \dots, K\}$  与单基站之间的单跳卸载机制。文献[19]建立多用户卸载至单 MAC 服务器系统模型,利用 OFDMA 技术将信道划分为  $N$  条子信道



if  $\widehat{L}(\widehat{i}) > 2\widehat{R}(\widehat{i})$  &  $\widehat{L}(\widehat{i}) \leq 2\widehat{R}(\widehat{i}) + \widehat{r}$

$$P_s(\widehat{i}) = 1 - \frac{\widehat{L}(\widehat{i}) - 2\widehat{R}(\widehat{i})}{\widehat{L}(\widehat{i})} \cdot \exp^{(-\rho_{\text{mob}}(2\widehat{R}(\widehat{i}) + 2\widehat{r} - \widehat{L}(\widehat{i})))};$$

else if  $\widehat{L}(\widehat{i}) > 2\widehat{R}(\widehat{i}) + \widehat{r}$  &  $\widehat{L}(\widehat{i}) \leq 2\widehat{R}(\widehat{i}) + 2\widehat{r}$

$$P_s(\widehat{i}) = 1 + \frac{2}{\rho_{\text{mob}}\widehat{L}(\widehat{i})} (\exp^{(-\rho_{\text{mob}}\widehat{r})} - \exp^{(-\rho_{\text{mob}}(2\widehat{R}(\widehat{i}) + 2\widehat{r} - \widehat{L}(\widehat{i})))}) - \frac{2\widehat{r} + 2\widehat{R}(\widehat{i}) - \widehat{L}(\widehat{i})}{\widehat{L}(\widehat{i})} \exp^{(-\rho_{\text{mob}}(2\widehat{R}(\widehat{i}) + 2\widehat{r} - \widehat{L}(\widehat{i})))};$$

else

$$P_s(\widehat{i}) = 1;$$

end

else if  $\widehat{L}(\widehat{i}) > 2\widehat{R}(\widehat{i}) + 2\widehat{r}$

$$P_s(\widehat{i}) = \frac{2\widehat{r} + 2\widehat{R}(\widehat{i})}{\widehat{L}(\widehat{i})} + \frac{2(\exp^{(-\rho_{\text{mob}}\widehat{r})} - 1)}{\rho_{\text{mob}}\widehat{L}(\widehat{i})};$$

else

$$P_s(\widehat{i}) = 1;$$

end

end

$[c, d] = \max(P_s);$   
 得出最优SBS

### 3.3 远海场景边缘计算卸载

图1另外描述了远海场景下多数据集实时并行执行场景，待处理数据可进行分割，分别在本地和附近 OECN 进行实时处理。与近海场景不同，远海场景网络连通时间有限，因此不能忽略 OECN 反馈结果时延。如图1所示，在同一时刻  $\bar{t}$ ， $\bar{K}$  个 OUE 将数据卸载至其通信范围内的 OECN。OUE  $\bar{k} \in \{1, 2, \dots, \bar{K}\}$  将待处理数据通过变量  $\mu_{\bar{k}}$  划分为部分本地处理和部分进行卸载处理。此时，利用 OFDMA 技术划分  $\bar{N}$  条子信道，可在  $\tau_{\bar{k}}$  足够大的情况下进一步缩短数据卸载时延，或在  $\tau_{\bar{k}}$  有限的情况下卸载更多数据。

## 4 近海计算卸载建模与算法设计

### 4.1 近海 MSU 建模与问题描述

在近海 MSU 模型中，OUE  $k$  的任务需求可用  $J_k = (D_{k, \text{Task}_i}, a_k, E_{k, \text{Task}_i, \text{max}})$ ,  $\forall k, \text{Task}_i$  表示。其中， $D_{k, \text{Task}_i}$  为 OUE  $k$  中  $\text{Task}_i$  的数据量（单位：bit）， $a_k$  为 OUE  $k$  的平均计算密度（单位：cycles/bit），

$E_{k, \text{Task}_i, \text{max}}$  为 OUE  $k$  允许  $\text{Task}_i$  的最大能耗（单位：J）。定义  $\text{fl}_k$  为 OUE  $k$  本地计算速率（单位：cycles/s）。利用 OFDMA 技术划分信道，由于子任务与子信道相互对应，将变量  $\mathbf{X} = \{x_{k, \text{Task}_i}, \forall k, \text{Task}_i\}$  定义为子任务卸载决策，同时也是子任务与子信道之间的配对决策。

在时刻  $T$ ，一个子任务最多只能选择一条子信道的卸载决策约束条件为

$$\sum_{k=1}^K \sum_{n=1}^N I(x_{k, \text{Task}_i}, n) \leq 1, \forall \text{Task}_i \quad (1)$$

一条子信道最多只能被一个子任务选择的卸载决策约束条件为

$$\sum_{k=1}^K \sum_{\text{Task}_i = \text{Task}_j}^{\text{Task}_s} I(x_{k, \text{Task}_i}, n) \leq 1, \forall n \quad (2)$$

其中， $I$  表示同或函数，即函数内的两个变量值相同，函数值为 1，反之为 0。对任意  $\forall k, \text{Task}_i$ ，若选择本地处理，则  $x_{k, \text{Task}_i} = 0$ 。本地处理时延  $t_{k, \text{Task}_i}^1$  及能耗  $E_{k, \text{Task}_i}^1$  分别记为

$$t_{k, \text{Task}_i}^1 = \frac{D_{k, \text{Task}_i} a_k}{\text{fl}_k}, \forall k, \text{Task}_i \quad (3)$$

$$E_{k, \text{Task}_i}^1 = l a_k \text{fl}_k^2 D_{k, \text{Task}_i}, \forall k, \text{Task}_i \quad (4)$$

其中， $l$  取决于处理器芯片结构的系数。

若选择卸载，则  $x_{k, \text{Task}_i}$  的取值为对应子信道标号，即  $x_{k, \text{Task}_i} = n, \forall n$ 。 $\text{Task}_i$  在子信道  $n$  上的上行链路传输速率为

$$r_{k, \text{Task}_i, n} = B_{k, \text{Task}_i, n} \text{lb} \left( 1 + \frac{P_{k, \text{Task}_i, n} g_{k, n}}{\sigma_n^2} \right), \forall k, \text{Task}_i, n \quad (5)$$

其中， $B_{k, \text{Task}_i, n}$ ,  $P_{k, \text{Task}_i, n}$  分别表示 OUE  $k$  中  $\text{Task}_i$  在子信道  $n$  上的带宽及传输功率， $\sigma_n^2$  表示子信道  $n$  上的高斯白噪声， $g_{k, n}$  表示 OUE  $k$  在子信道  $n$  上的信道增益。

因此， $\text{Task}_i$  在子信道  $n$  上的数据传输时延和能耗可分别表示为

$$t_{k, \text{Task}_i}^r = \frac{D_{k, \text{Task}_i}}{r_{k, \text{Task}_i, n}}, \forall k, \text{Task}_i, n \quad (6)$$

$$E_{k, \text{Task}_i}^o = p_{k, \text{Task}_i, n} t_{k, \text{Task}_i}^r, \forall k, \text{Task}_i, n \quad (7)$$

SBS 处理时延可表示为

$$t_{k, \text{Task}_i}^c = \frac{D_{k, \text{Task}_i} a_k}{f_c}, \forall k, \text{Task}_i \quad (8)$$

其中,  $f_c$  表示 SBS 的计算速率。卸载总时延为

$$t_{k, \text{Task}_i}^o = t_{k, \text{Task}_i}^r + t_{k, \text{Task}_i}^c \quad (9)$$

为了缩短每个 OUE 的时延, 着手于每个子任务, 联合考虑其本地处理时延和卸载总时延, 并将两者的差值作为优化目标, 即  $t_{\text{fit}, k, \text{Task}_i} = t_{k, \text{Task}_i}^l - t_{k, \text{Task}_i}^o, \forall k, \text{Task}_i$ 。同时, 为了保障各子任务之间的公平性, 将上述优化目标  $t_{\text{fit}, k, \text{Task}_i}$  以每个 OUE 为单位, 求和取平均作为优化目标函数。基于上述讨论, 优化问题可表示为

$$\max_{\{\mathbf{X}, \mathbf{P}\}} T_{\text{fit}, k}(\mathbf{X}, \mathbf{P}), \forall k$$

$$\text{s.t. C1: } x_{k, \text{Task}_i} \in \{0, 1, \dots, N\}, \forall k, \text{Task}_i$$

$$\text{C2: } \sum_{k=1}^K \sum_{\text{Task}_i=\text{Task}_1}^{\text{Task}_S} I(x_{k, \text{Task}_i}, n) \leq 1, \forall k, n$$

$$\text{C3: } \sum_{k=1}^K \sum_{n=1}^N I(x_{k, \text{Task}_i}, n) \leq 1, \forall k, \text{Task}_i$$

$$\text{C4: } I(x_{k, \text{Task}_i}, 0) E_{k, \text{Task}_i}^l + I(x_{k, \text{Task}_i}, n) E_{k, \text{Task}_i}^o \leq E_{k, \text{Task}_i, \max}, \forall k, \text{Task}_i, n$$

$$\text{C5: } 0 \leq p_{k, \text{Task}_i, n} \leq p_{k, \text{Task}_i, \max}, \forall k, \text{Task}_i, n \quad (10)$$

其中,  $T_{\text{fit}, k}(\mathbf{X}, \mathbf{P}) = \frac{\sum_{\text{Task}_i=\text{Task}_1}^{\text{Task}_S} I(x_{k, \text{Task}_i}, n) t_{\text{fit}, k, \text{Task}_i}}{S}, \forall k, n$ ,

$\mathbf{P} = \{p_{k, \text{Task}_i, x_{k, \text{Task}_i}}, \forall k, \text{Task}_i\}$ 。约束 C1 为卸载决策的取值范围; 约束 C2、C3 保证子任务与子信道是相互对应的关系; 约束 C4 保证每个子任务实际能耗不超过其最大能耗; 约束 C5 保证每个子任务的传输功率不超过其额定功率。

## 4.2 近海 MSU 卸载算法描述

式(10)为混合整数非线性规划问题, 包含二进制卸载决策  $\mathbf{X}$ 、连续传输功率  $\mathbf{P}$  两个变量, 将上述问题划分成传输功率分配优化和卸载决策优化两个子问题并分别求解。

### 4.2.1 传输功率分配优化

假设所有子任务均卸载处理, 即  $x_{k, \text{Task}_i} = n, \forall k, \text{Task}_i, n$ , 最优传输功率只与传输时延有关, 简化式(10)得

$$\min_{\{\mathbf{P}\}} t_{k, \text{Task}_i}^r, \forall k, \text{Task}_i, n$$

$$\text{s.t. C6: } E_{k, \text{Task}_i}^o \leq E_{k, \text{Task}_i, \max}, \forall k, \text{Task}_i, n \quad (11)$$

C5

$$\text{令 } \lambda_{k, \text{Task}_i} = \frac{1}{t_{k, \text{Task}_i}^r}, \forall k, \text{Task}_i, \text{ 得}$$

$$\max \lambda_{k, \text{Task}_i}, \forall k, \text{Task}_i$$

$$\text{s.t. C7: } H_{k, \text{Task}_i}(\lambda_{k, \text{Task}_i}) \geq 0, \forall k, \text{Task}_i \quad (12)$$

$$\text{C8: } \lambda_{k, \text{Task}_i} \leq \lambda_{k, \text{Task}_i, \max}, \forall k, \text{Task}_i$$

其中,  $H_{k, \text{Task}_i}(\lambda_{k, \text{Task}_i}) = E_{k, \text{Task}_i, \max} \lambda_{k, \text{Task}_i} - \frac{\sigma_n^2}{g_{k, n}}$

$$\left( 2^{\frac{D_{k, \text{Task}_i}}{B_{k, \text{Task}_i, n}} \lambda_{k, \text{Task}_i}} - 1 \right), \lambda_{k, \text{Task}_i, \max} = \frac{B_{k, \text{Task}_i, n}}{D_{k, \text{Task}_i}} \text{lb} \left( 1 + \frac{p_{k, \text{Task}_i, \max} g_{k, n}}{\sigma_n^2} \right),$$

$\forall k, \text{Task}_i, n$ 。

对  $H_{k, \text{Task}_i}(\lambda_{k, \text{Task}_i})$  求得

$$H'_{k, \text{Task}_i}(\lambda_{k, \text{Task}_i}) = E_{k, \text{Task}_i, \max} - \frac{\sigma_n^2 2^{\frac{D_{k, \text{Task}_i}}{B_{k, \text{Task}_i, n}} \lambda_{k, \text{Task}_i}} \ln 2 D_{k, \text{Task}_i}}{g_{k, n} B_{k, \text{Task}_i, n}}, \quad \forall k, \text{Task}_i, n \quad (13)$$

由式(13)可知,  $H_{k, \text{Task}_i}(\lambda_{k, \text{Task}_i})$  是一个存在极大值的单峰函数, 则式(12)目标函数最优解为  $H_{k, \text{Task}_i}(\lambda_{k, \text{Task}_i})$  的非 0 点与  $\lambda_{k, \text{Task}_i, \max}$  之间的较小值。 $H_{k, \text{Task}_i}(\lambda_{k, \text{Task}_i})=0$  难以直接求解, 文献[21]利用二分搜索法无限逼近最优值。但对于每个 OUE 的每个子任务到各子信道的传输功率而言, 采用二分搜索法会增加计算复杂度与时延。考虑  $H_{k, \text{Task}_i}(\lambda_{k, \text{Task}_i})$  的极值点容易求得并且满足式(12)约束 C7, 于是次优解为  $\lambda_{k, \text{Task}_i}^* = \max(0, \min(\lambda_{k, \text{Task}_i, \max}, \lambda_{k, \text{Task}_i, \max}))$

$$\lambda_{k, \text{Task}_i, \max} = \frac{B_{k, \text{Task}_i, n} \text{lb} \frac{B_{k, \text{Task}_i, n} g_{k, n} E_{k, \text{Task}_i, \max}}{\sigma_n^2 \ln 2 D_{k, \text{Task}_i}}}{D_{k, \text{Task}_i}}, \forall k, \text{Task}_i, n \quad (14)$$

从而得出该条件下的传输功率分配表达式为

$$p_{k, \text{Task}_i, n}^* = \frac{\sigma_n^2}{g_{k, n}} \left( 2^{\frac{D_{k, \text{Task}_i}}{B_{k, \text{Task}_i, n}} \lambda_{k, \text{Task}_i}^*} - 1 \right), \forall k, \text{Task}_i, n \quad (15)$$

### 4.2.2 卸载决策优化

解决传输功率分配优化问题后, 得出仅含有卸载决策的优化问题。

$$\begin{aligned} & \max_{\{X\}} T_{\text{fit},k}(\mathbf{X}, \mathbf{P}^*), \forall k \\ \text{s.t. } & \text{C9: } I(x_{k,\text{Task}_i}, 0)E_{k,\text{Task}_i}^1 + I(x_{k,\text{Task}_i}, n)E_{k,\text{Task}_i}^{0*} \leq (16) \\ & E_{k,\text{Task}_i,\text{max}}, \forall k, \text{Task}_i, n \\ & \text{C1, C2, C3} \end{aligned}$$

其中,  $E_{k,\text{Task}_i}^{0*}$  为代入  $p_{k,\text{Task}_i,x_{k,\text{Task}_i}}^*$  后的卸载传输能耗。

实现多 OUE 中多子任务与多子信道之间的优化配对决策复杂度较高, 为满足 OUE 低时延应用需求, 应综合考虑每个 OUE 本地处理时延及卸载传输时延, 并将两者决定变量 (不包含以优先级函数划分结果为目的的信道资源分配变量) 整合到一个表达式作为优先级函数, 以确定 OUE 信道资源分配顺序并降低系统复杂度。卸载优先级函数可表示为

$$\text{pr}_k = \frac{\text{lb}\left(1 + \sum_{\text{Task}_j=\text{Task}_1}^{\text{Task}_S} \sum_{n=1}^N p_{k,\text{Task}_j,n}^*\right) a_k}{\text{fl}_k}, \forall k \quad (17)$$

该函数由  $\text{fl}$ 、 $a$ 、 $p^*$  3 个变量组成。 $\text{fl}$  增大或  $a$  减小则本地处理时延减小, OUE 卸载优先级降低;  $p^*$  增大时, 在满足能耗约束的情况下可节省任务卸载时延, OUE 卸载优先级增高。

根据式(17), 确定各 OUE 卸载优先级后, 利用 AFSA 求解每个 OUE 卸载决策问题<sup>[19]</sup>。AFSA 通过构造人工鱼模仿鱼群的觅食、聚群及追尾行为实现寻优。AFSA 参数与近海 MSU 卸载决策优化参数间的映射关系如表 1 所示。结合任务卸载, 表 1 中人工鱼的位置可表示为  $\mathbf{X}_k = \{x_{\text{Task}_1}, x_{\text{Task}_2}, \dots, x_{\text{Task}_S}\}, \forall k$ , 且位置向量的维度与子任务数  $S$  一致, 位置向量的解空间可表示为  $x_{\text{Task}_i} \in \{0, 1, 2, \dots, N\}, \forall \text{Task}_i$ 。对于适应度函数, 引入子信道带宽均分策略解决不满足式(10)约束 C2、C3 的情况, 引入惩罚因子解决不满足式(10)约束 C4 的情况, 则适应度函数可表示为

$$\begin{aligned} \text{Fitness}(\mathbf{X}_k) = & \max_{\{X_k\}} \left( \sum_{\text{Task}_j=\text{Task}_1}^{\text{Task}_S} I(x_{k,\text{Task}_j}, n) \left( (t_{k,\text{Task}_j}^1 - t_{k,\text{Task}_j}^{0*}) - \right. \right. \\ & \left. \left. g_1 \max(E_{k,\text{Task}_j}^{0*} - E_{k,\text{Task}_j,\text{max}}, 0) \right) \right) / S, \forall k \quad (18) \end{aligned}$$

其中,  $t_{k,\text{Task}_j}^{0*}$  为代入  $p_{k,\text{Task}_j,x_{k,\text{Task}_j}}^*$  后的卸载总时延。

式(18)由两部分组成。从 OUE $k$  的单个  $\text{Task}_i$  着手, 第一部分为本地处理时延与卸载总时延 (代入

$p_{k,\text{Task}_j,x_{k,\text{Task}_j}}^*$ ) 的差值; 第二部分为其引入惩罚因子满足能耗约束。将上述两部分以差的形式联立, 并以 OUE  $k$  为单位求和取平均, 则可以构成近海 MSU 模型下的适应度函数。显然, 该适应度函数值越大, OUE  $k$  卸载处理节省时延效果越好。

表 1 AFSA 参数与近海 MSU 卸载决策优化参数间的映射关系

AFSA	近海 MSU 卸载决策优化问题
人工鱼的位置	卸载决策向量
解空间	卸载决策的取值范围
适应度函数	优化目标函数

为了不影响信道资源分配效率并获得最优卸载决策, 需要提前分离 OUE 中部分不可卸载的子任务。确定 OUE 任务卸载优先级后, 在传统人工鱼群算法基础上, 本文提出基于变异操作的人工鱼群算法 (MO-AFSA, artificial fish swarm algorithm based on mutation operation), 可使得在基本参数 (人工鱼数量、最大尝试次数、最大迭代次数等) 较小的情况下跳出局部最优。MO-AFSA 具体改进包括: 1) 将人工鱼寻优的固定步长变为可变步长; 2) 在迭代过程中, 在每条人工鱼分别进行觅食、聚群和追尾后, 对 3 种行为下的位置坐标解进行二进制编码并以一定概率发生变异, 判断变异后的解能否取得更优值<sup>[19]</sup>。所提近海 MSU 卸载方案下的 MO-AFSA 伪代码如算法 2 所示。

**算法 2** 近海 MSU 卸载方案下的 MO-AFSA 伪代码

```

输入  $K, S$ , 人工鱼数量 fishNum, 最大尝试次数 tryNum
拥挤度因子  $\delta$ , 最大迭代次数 MAX_GEN
输出  $\mathbf{X}$ , 最优 Fitness
for  $k = 1 : K$ 
    分离 OUE  $k$  不可卸载子任务
    更新 OUE  $k$  可卸载子任务数量  $S_k$ 
    根据式(17) 计算  $\text{pr}_k$ 
end
 $[\sim, m] = \text{sort}(\text{pr}, 'descend');$ 
更新各 OUE 卸载顺序
for  $k = m(1) : m(K)$ 
    随机初始化 fishNum 条人工鱼的位置,
    迭代次数  $it = 0$ 
    
```

```

while it < MAX_GEN do
  it+ = 1
  for j = 1: fishNum do
    第j条人工鱼执行觅食行为后进行
    变异操作如算法3
    第j条人工鱼执行聚群行为后进行
    变异操作如算法3
    第j条人工鱼执行追尾行为后进行
    变异操作如算法3
    以上3个行为中, 返回值最优的
    作为第j条
    人工鱼的下一个位置
  end
end while
return 人工鱼群中鱼的最优位置
end

```

### 4.3 近海 MMU 卸载算法描述

在 MMU 模型中, OUE  $\tilde{k}$  的任务需求可用  $\tilde{J}_{\tilde{k}}(\tilde{D}_{\tilde{k}, \tilde{\text{Task}}_i}, \tilde{a}_{\tilde{k}}, \tilde{E}_{\tilde{k}, \tilde{\text{Task}}_i}^{\max}), \forall \tilde{k}, \tilde{\text{Task}}_i$  表示。  $\tilde{t}_{\tilde{k}, \tilde{\text{Task}}_i}^r$ 、 $\tilde{E}_{\tilde{k}, \tilde{\text{Task}}_i}^o$  分别表示 OUE  $\tilde{k}$  中  $\tilde{\text{Task}}_i$  通过子信道  $\tilde{n}$  传输至一级 OECN 的时延与能耗, 具体表达式见式(6)和式(7)。与 MSU 模型不同, 由于卸载节点多样, MMU 卸载决策  $\tilde{X} = \{\tilde{x}_{\tilde{k}, \tilde{\text{Task}}_i}, \forall \tilde{k}, \tilde{\text{Task}}_i\}$  仅能通过子任务与子信道之间的配对决策判断是否卸载或利用哪一条子信道完成卸载。定义变量  $\tilde{Y} = \{\tilde{y}_{\tilde{k}, \tilde{\text{Task}}_i}, \forall \tilde{k}, \tilde{\text{Task}}_i\}$ ,  $\tilde{y}_{\tilde{k}, \tilde{\text{Task}}_i} = 1$  表示任务卸载到一级 OECN,  $\tilde{y}_{\tilde{k}, \tilde{\text{Task}}_i} = 2$  表示任务卸载到二级 OECN。  $\forall \tilde{k}, \tilde{\text{Task}}_i$ , 若  $\tilde{\text{Task}}_i$  本地处理, 则  $\tilde{x}_{\tilde{k}, \tilde{\text{Task}}_i} = 0$ 。此时, 时延  $\tilde{t}_{\tilde{k}, \tilde{\text{Task}}_i}^1$  和能耗  $\tilde{E}_{\tilde{k}, \tilde{\text{Task}}_i}^1$  可用式(3)和式(4)表示。若  $\tilde{\text{Task}}_i$  由一级 OECN 处理, 则  $\tilde{y}_{\tilde{k}, \tilde{\text{Task}}_i} = 1, \tilde{x}_{\tilde{k}, \tilde{\text{Task}}_i} \in \{1, 2, \dots, \tilde{N}_{\tilde{k}}\}$ , 节点处理时延为  $\tilde{t}_{\tilde{k}, \tilde{\text{Task}}_i}^m = \frac{\tilde{D}_{\tilde{k}, \tilde{\text{Task}}_i} \tilde{a}_{\tilde{k}}}{\tilde{f}_{\tilde{k}, \tilde{\text{Task}}_i}^m}$ 。其中,  $\tilde{f}_{\tilde{k}, \tilde{\text{Task}}_i}^m$  表示 OUE  $\tilde{k}$  中  $\tilde{\text{Task}}_i$  所分得的一级 OECN 的计算速率。

由于一级 OECN 计算资源不足, 各由一级 OECN 处理的子任务需依据先到先得策略在该一级 OECN 排队等候处理。假设各子任务在一级 OECN 处理顺序为  $q_{\tilde{\text{Task}}_1}, q_{\tilde{\text{Task}}_2}, \dots, q_{\tilde{\text{Task}}_j}, \dots, q_{\tilde{\text{Task}}_J}, \tilde{j} \leq \tilde{S}_{\tilde{k}}, \forall \tilde{k}$ , 则  $\tilde{\text{Task}}_i$  在一级 OECN 的排队等候时延为

$$\text{tw}_{\tilde{k}, \tilde{\text{Task}}_i} = \begin{cases} 0, \tilde{\text{Task}}_i = \tilde{\text{Task}}_1 \\ \max\left(0, \tilde{t}_{\tilde{k}, \tilde{\text{Task}}_i}^m + \tilde{t}_{\tilde{k}, \tilde{\text{Task}}_i}^r - \tilde{t}_{\tilde{k}, \tilde{\text{Task}}_i}^r\right), \tilde{\text{Task}}_i \neq \tilde{\text{Task}}_1 \end{cases}$$

$\forall \tilde{k}, \tilde{\text{Task}}_i$ ; 若  $\tilde{\text{Task}}_i$  由二级 OECN 处理, 则  $\tilde{y}_{\tilde{k}, \tilde{\text{Task}}_i} = 2, \tilde{x}_{\tilde{k}, \tilde{\text{Task}}_i} \in \{1, 2, \dots, \tilde{N}_{\tilde{k}}\}$ , 二级 OECN 处理时延  $\tilde{t}_{\tilde{k}, \tilde{\text{Task}}_i}^o$  与式(8)一致。

考虑近海节点资源丰富, 为有效缩短卸载总时延, 该模型规定 OUE 与一级 OECN、子任务与子信道均满足一一对应关系, 其约束条件可表示为

$$\begin{aligned} & \sum_{\tilde{n}=1}^{\tilde{N}_{\tilde{k}}} I(\tilde{y}_{\tilde{k}, \tilde{\text{Task}}_i}, 1) I(\tilde{x}_{\tilde{k}, \tilde{\text{Task}}_i}, \tilde{n}) + \\ & I(\tilde{y}_{\tilde{k}, \tilde{\text{Task}}_i}, 2) I(\tilde{x}_{\tilde{k}, \tilde{\text{Task}}_i}, \tilde{n}) \leq 1, \forall \tilde{k}, \tilde{\text{Task}}_i \quad (19) \\ & \sum_{\tilde{\text{Task}}_j = \tilde{\text{Task}}_1}^{\tilde{\text{Task}}_{\tilde{S}_{\tilde{k}}}} I(\tilde{y}_{\tilde{k}, \tilde{\text{Task}}_j}, 1) I(\tilde{x}_{\tilde{k}, \tilde{\text{Task}}_j}, \tilde{n}) + \\ & I(\tilde{y}_{\tilde{k}, \tilde{\text{Task}}_j}, 2) I(\tilde{x}_{\tilde{k}, \tilde{\text{Task}}_j}, \tilde{n}) \leq 1, \forall \tilde{k}, \tilde{n} \quad (20) \end{aligned}$$

本节以最小化每个 OUE  $\tilde{k}$  的总时延(即 OUE  $\tilde{k}$  的所有子任务中最大时延)作为优化目标。基于上述讨论, 优化问题可表示为

$$\begin{aligned} & \min_{\{\tilde{X}, \tilde{Y}, \tilde{P}\}} \tilde{T}_{\text{fit}, \tilde{k}}(\tilde{X}, \tilde{Y}, \tilde{P}), \forall \tilde{k}, \tilde{\text{Task}}_i, \tilde{n} \\ & \text{s.t. C1: } \tilde{x}_{\tilde{k}, \tilde{\text{Task}}_i} \in \{0, 1, \dots, \tilde{N}_{\tilde{k}}\}, \forall \tilde{k}, \tilde{\text{Task}}_i \\ & \text{C2: } \tilde{y}_{\tilde{k}, \tilde{\text{Task}}_i} \in \{1, 2\}, \forall \tilde{k}, \tilde{\text{Task}}_i \\ & \text{C3: } \sum_{\tilde{n}=1}^{\tilde{N}_{\tilde{k}}} I(\tilde{y}_{\tilde{k}, \tilde{\text{Task}}_i}, 1) I(\tilde{x}_{\tilde{k}, \tilde{\text{Task}}_i}, \tilde{n}) + \\ & I(\tilde{y}_{\tilde{k}, \tilde{\text{Task}}_i}, 2) I(\tilde{x}_{\tilde{k}, \tilde{\text{Task}}_i}, \tilde{n}) \leq 1, \forall \tilde{k}, \tilde{\text{Task}}_i \\ & \text{C4: } \sum_{\tilde{\text{Task}}_j = \tilde{\text{Task}}_1}^{\tilde{\text{Task}}_{\tilde{S}_{\tilde{k}}}} I(\tilde{y}_{\tilde{k}, \tilde{\text{Task}}_j}, 1) I(\tilde{x}_{\tilde{k}, \tilde{\text{Task}}_j}, \tilde{n}) + \\ & I(\tilde{y}_{\tilde{k}, \tilde{\text{Task}}_j}, 2) I(\tilde{x}_{\tilde{k}, \tilde{\text{Task}}_j}, \tilde{n}) \leq 1, \forall \tilde{k}, \tilde{n} \\ & \text{C5: } I(\tilde{x}_{\tilde{k}, \tilde{\text{Task}}_i}, 0) \tilde{E}_{\tilde{k}, \tilde{\text{Task}}_i}^1 + \\ & I(\tilde{x}_{\tilde{k}, \tilde{\text{Task}}_i}, \tilde{n}) \tilde{E}_{\tilde{k}, \tilde{\text{Task}}_i}^o \leq \tilde{E}_{\tilde{k}, \tilde{\text{Task}}_i}^{\max}, \forall \tilde{k}, \tilde{\text{Task}}_i, \tilde{n} \\ & \text{C6: } 0 \leq \tilde{p}_{\tilde{k}, \tilde{\text{Task}}_i, \tilde{n}} \leq \tilde{p}_{\tilde{k}, \tilde{\text{Task}}_i}^{\max}, \forall \tilde{k}, \tilde{\text{Task}}_i, \tilde{n} \end{aligned} \quad (21)$$

其中,  $\tilde{T}_{\text{fit}, \tilde{k}}(\tilde{X}, \tilde{Y}, \tilde{P}) = \max \left( \sum_{\tilde{\text{Task}}_j = \tilde{\text{Task}}_1}^{\tilde{\text{Task}}_J} I(\tilde{x}_{\tilde{k}, \tilde{\text{Task}}_j}, 0) \tilde{t}_{\tilde{k}, \tilde{\text{Task}}_j}^1, \right.$

$$I(\tilde{x}_{\tilde{k}, \tilde{\text{Task}}_i}, \tilde{n}) I(\tilde{y}_{\tilde{k}, \tilde{\text{Task}}_i}, 1) \tilde{t}_{\tilde{k}, \tilde{\text{Task}}_i}^{o1}, I(\tilde{x}_{\tilde{k}, \tilde{\text{Task}}_i}, \tilde{n}) I(\tilde{y}_{\tilde{k}, \tilde{\text{Task}}_i}, 2) \tilde{t}_{\tilde{k}, \tilde{\text{Task}}_i}^{o2},$$

$$\left. \forall \tilde{\text{Task}}_i, \tilde{n}, \forall \tilde{k}, \tilde{t}_{\tilde{k}, \tilde{\text{Task}}_i}^{o1} = \tilde{t}_{\tilde{k}, \tilde{\text{Task}}_i}^r + \tilde{t}_{\tilde{k}, \tilde{\text{Task}}_i}^m + \text{tw}_{\tilde{k}, \tilde{\text{Task}}_i}, \forall \tilde{k}, \right.$$

$\tilde{\text{Task}}_i$  为一级卸载总时延,  $\tilde{t}_{\tilde{k}, \tilde{\text{Task}}_i}^{o2} = \tilde{t}_{\tilde{k}, \tilde{\text{Task}}_i}^r + \tilde{t}_{\tilde{k}, \tilde{\text{Task}}_i}^c + \text{tts}_{\tilde{k}, \tilde{\text{Task}}_i}, \forall \tilde{k}, \tilde{\text{Task}}_i$  二级卸载总时延。  $\text{tts}_{\tilde{k}, \tilde{\text{Task}}_i}$  表示 OUE  $\tilde{k}$  中  $\tilde{\text{Task}}_i$  由中继节点传输至 SBS 的传输时延。

式(21)约束 C1、C2 为子任务卸载决策取值; 约束 C3 保证一个子任务最多只能选择一条子信道; 约束 C4 保证一条子信道最多只能服务于一个子任务; 约束 C5 保证每个子任务实际能耗不超过其最大能耗; 约束 C6 保证每个子任务的传输功率不超过其额定功率。

#### 4.4 卸载决策同步优化

MMU 传输功率分配优化问题与 MSU 问题一致, 此处不再阐述。在传输功率分配优化基础上, 需要优化的目标函数及约束条件可表示为

$$\begin{aligned} & \min_{\{\tilde{\mathbf{X}}, \tilde{\mathbf{Y}}\}} \tilde{T}_{\tilde{k}}(\tilde{\mathbf{X}}, \tilde{\mathbf{Y}}, \tilde{\mathbf{P}}^*) \\ \text{s.t. } & \text{C7: } I(\tilde{x}_{\tilde{k}, \tilde{\text{Task}}_i}, 0) \tilde{E}_{\tilde{k}, \tilde{\text{Task}}_i}^1 + \\ & I(\tilde{x}_{\tilde{k}, \tilde{\text{Task}}_i}, \tilde{n}) \tilde{E}_{\tilde{k}, \tilde{\text{Task}}_i}^{o*} \leq \tilde{E}_{\tilde{k}, \tilde{\text{Task}}_i, \max}, \forall \tilde{k}, \tilde{\text{Task}}_i, \tilde{n} \\ & \text{C1, C2, C3, C4} \end{aligned} \quad (22)$$

其中,  $\tilde{E}_{\tilde{k}, \tilde{\text{Task}}_i}^{o*}$  为代入  $\tilde{p}_{\tilde{k}, \tilde{\text{Task}}_i, \tilde{x}_{\tilde{k}, \tilde{\text{Task}}_i}}^*$  后的卸载传输能耗。

在同一时刻  $\tilde{T}$ , 通信范围内每个 OUE 与计算资源差异明显的一级 OECN 均可互通。为保证 OUE 之间的公平性, 提高时延敏感型 OUE 的体验品质, 综合考虑每个 OUE 本地处理时延及卸载传输时延, 并将其决定变量(本节针对一级 OECN 选取问题划分优先级, 因此包含信道资源变量)整合成一个表达式作为判断 OUE 选取一级 OECN 的顺序(优先级高的 OUE 优先选取计算资源较为充足的一级卸载或中继节点)。优先级函数可表示为

$$\text{prl}_{\tilde{k}} = \frac{\sum_{\tilde{\text{Task}}_i = \text{Task}_1}^{\tilde{\text{Task}}_{S_{\tilde{k}}}} \sum_{\tilde{n}=1}^{\tilde{N}_{\tilde{k}}} \tilde{B}_{\tilde{k}, \tilde{\text{Task}}_i, \tilde{n}} \text{lb} \left( 1 + \frac{\sum_{\tilde{\text{Task}}_i = \text{Task}_1}^{\tilde{\text{Task}}_{S_{\tilde{k}}}} \sum_{\tilde{n}=1}^{\tilde{N}_{\tilde{k}}} \tilde{P}_{\tilde{k}, \tilde{\text{Task}}_i, \tilde{n}}^* \tilde{\sigma}_{\tilde{n}}^2}{\tilde{\sigma}_{\tilde{n}}^2} \right) \tilde{a}_{\tilde{k}}}{\tilde{f}_{\tilde{k}}}, \forall \tilde{k} \quad (23)$$

利用 MO-AFSA 同时优化两个卸载决策问题, 算法与优化问题参数间的映射关系参照表 1。结合卸载场景, 表 1 中人工鱼的位置为

$$\begin{pmatrix} \tilde{\mathbf{X}}_{\tilde{k}} \\ \tilde{\mathbf{Y}}_{\tilde{k}} \end{pmatrix} = \begin{bmatrix} \tilde{x}_{\tilde{\text{Task}}_1}, \tilde{x}_{\tilde{\text{Task}}_2}, \dots, \tilde{x}_{\tilde{\text{Task}}_{S_{\tilde{k}}}} \\ \tilde{y}_{\tilde{\text{Task}}_1}, \tilde{y}_{\tilde{\text{Task}}_2}, \dots, \tilde{y}_{\tilde{\text{Task}}_{S_{\tilde{k}}}} \end{bmatrix}, \forall \tilde{k}, \text{ 位置向量解空}$$

间为  $\tilde{x}_{\tilde{\text{Task}}_i} \in \{0, 1, 2, \dots, \tilde{N}_{\tilde{k}}\}$ ,  $\tilde{y}_{\tilde{\text{Task}}_i} \in \{1, 2\}, \forall \tilde{k}, \tilde{\text{Task}}_i$ 。

与 MSU 模型策略一致, 适应度函数可表示为

$$\begin{aligned} & \text{Fitness1}(\tilde{\mathbf{X}}_{\tilde{k}}, \tilde{\mathbf{Y}}_{\tilde{k}}) = \min_{\{\tilde{\mathbf{X}}_{\tilde{k}}, \tilde{\mathbf{Y}}_{\tilde{k}}\}} \\ & \max \left( \sum_{\tilde{\text{Task}}_i = \text{Task}_1}^{\tilde{\text{Task}}_{S_{\tilde{k}}}} I(\tilde{x}_{\tilde{k}, \tilde{\text{Task}}_i}, 0) \tilde{t}_{\tilde{k}, \tilde{\text{Task}}_i}^1, I(\tilde{y}_{\tilde{k}, \tilde{\text{Task}}_i}, 1) \cdot \right. \\ & I(\tilde{x}_{\tilde{k}, \tilde{\text{Task}}_i}, \tilde{n}) \tilde{t}_{\tilde{k}, \tilde{\text{Task}}_i}^{o1*}, I(\tilde{y}_{\tilde{k}, \tilde{\text{Task}}_i}, 2) I(\tilde{x}_{\tilde{k}, \tilde{\text{Task}}_i}, \tilde{n}) \tilde{t}_{\tilde{k}, \tilde{\text{Task}}_i}^{o2*} + \\ & \left. \tilde{g}1 \max \left( I(\tilde{x}_{\tilde{k}, \tilde{\text{Task}}_i}, 0) \tilde{E}_{\tilde{k}, \tilde{\text{Task}}_i}^1 + I(\tilde{x}_{\tilde{k}, \tilde{\text{Task}}_i}, \tilde{n}) \tilde{E}_{\tilde{k}, \tilde{\text{Task}}_i}^{o*} - \tilde{E}_{\tilde{k}, \tilde{\text{Task}}_i, \max}, 0 \right), \right. \\ & \left. \forall \tilde{\text{Task}}_i, \tilde{n}, \forall \tilde{k} \right) \end{aligned} \quad (24)$$

近海 MMU 卸载方案下的 MO-AFSA 伪代码如下算法 3 所示。

#### 算法 3 变异操作伪代码

输入 人工鱼  $j$  的位置坐标  $\mathbf{X}_j$ , 变异概率  $\text{pm}$ , 位置坐标维度  $S_j$

(仅包含可卸载部分子任务), 编码位数  $\text{bi}$ , 划分子信道总数  $N$

输出 进行变异操作之后人工鱼  $j$  的位置坐标  $\mathbf{X}_j^m$

令人工鱼  $j$  的位置坐标  $\mathbf{X}_j$  内各元素区间范围

变换为  $[0, 2^{\text{bi}} - 1]$

即  $\mathbf{X}_j = \text{round}(\mathbf{X}_j (2^{\text{bi}} - 1) / N)$ ;

将  $\mathbf{X}_j$  进行二进制编码(编码  $\text{bi}$  位)

$\text{pop} = \text{dec2bin}(\mathbf{X}_j, \text{bi})$ ;

for  $\text{Task}_i = \text{Task}_1 : \text{Task}_{S_j}$

if ( $\text{rand} < \text{pm}$ )

$\text{mpoint} = \text{round}(\text{rand} * \text{bi})$ ;

    if  $\text{mpoint} \leq 0$ ;

$\text{mpoint} = 1$ ;

    end

$\text{newpop}(\text{Task}_i, :) = \text{pop}(\text{Task}_i, :)$ ;

        if  $\text{newpop}(\text{Task}_i, \text{mpoint}) = 0$

$\text{newpop}(\text{Task}_i, \text{mpoint}) = 1$ ;

        else  $\text{newpop}(\text{Task}_i, \text{mpoint}) = 1$ ;

$\text{newpop}(\text{Task}_i, \text{mpoint}) = 0$ ;

    end

else  $\text{newpop}(\text{Task}_i, :) = \text{pop}(\text{Task}_i, :)$ ;

end

end

$$X_j^m = \text{round}\left(\frac{\text{bin2dec}(\text{newpop}) * N}{2^{\text{bi}} - 1}\right);$$

首先, 通过式(23)确定各 OUE 选择一级卸载或中继节点优先级 (第 1 行—第 4 行)。其次, 令每条人工鱼在进行觅食行为、聚群行为和追尾行为后以一定的概率进行变异操作 (具体过程如算法 4 所示), 不断尝试跳出局部最优 (第 5 行~第 16 行)。

**算法 4** 近海 MMU 卸载方案下的 MO-AFSA 伪代码

输入  $\tilde{K}, \tilde{S}_k$ , 人工鱼数量 fishNum1, 最大尝试次数 tryNum1, 变异概率 pm1, 拥挤度因子  $\delta I$ , 最大迭代次数 MAX\_

GEN1

输出 最优位置, 最优 Fitness1

for  $\tilde{k} = 1: \tilde{K}$

根据式 (23) 计算  $\text{prl}_{\tilde{k}}$

end

$[\sim, \tilde{m}] = \text{sort}(\text{prl}, 'descend')$  得出各个 OUE 的优先级

for  $\tilde{k} = \tilde{m}(1): \tilde{m}(\tilde{K})$

随机初始化 fishNum1 条人工鱼的位置,  
迭代次数 it = 0

while it < MAX\_GEN1 do

it += 1

for  $\tilde{i} = 1: \text{fishNum1}$  do

第  $\tilde{i}$  条人工鱼执行觅食行为后进行变异操作如算法 3

第  $\tilde{i}$  条人工鱼执行聚群行为后进行变异操作如算法 3

第  $\tilde{i}$  条人工鱼执行追尾行为后进行变异操作如算法 3

以上 3 个行为中, 返回值最优的作为第  $\tilde{i}$  条人工鱼的下一个位置

end while

return 人工鱼群中鱼的最优位置

end

## 5 远海计算卸载建模与算法设计

### 5.1 远海计算卸载建模与问题描述

在远海 MSU 模型中, OUE  $\bar{k}$  的任务需求可用  $\bar{J}_{\bar{k}} = (\bar{D}_{\bar{k}}, \bar{a}_{\bar{k}}, \bar{E}_{\bar{k}_{\max}})$  表示。对于任意 OUE  $\bar{k}$ ,  $\bar{D}_{\bar{k}}$  表

示待处理数据总量 (单位: bit),  $\bar{a}_{\bar{k}}$  为其平均计算密度 (单位: cycles/bit),  $\bar{E}_{\bar{k}_{\max}}$  为其所允许的最大能耗 (单位: J)。设  $f_{\bar{k}}$  为本地计算速率 (单位: cycles/s),  $\mu_{\bar{k}}$  为其卸载数据量与其待处理的数据总量之比, 则 OUE  $\bar{k}$  本地处理总时延  $\bar{t}_{\bar{k}}^l$  与本地能耗  $\bar{E}_{\bar{k}}^l$  可分别表示为

$$\bar{t}_{\bar{k}}^l = \frac{\bar{D}_{\bar{k}} \bar{a}_{\bar{k}} (1 - \mu_{\bar{k}})}{f_{\bar{k}}}, \forall \bar{k} \quad (25)$$

$$\bar{E}_{\bar{k}}^l = l \bar{D}_{\bar{k}} \bar{a}_{\bar{k}} (1 - \mu_{\bar{k}}) f_{\bar{k}}^2, \forall \bar{k} \quad (26)$$

其中,  $l$  为取决于芯片结构的系数。

在远海系统模型中, 规定每条子信道带宽相同并设为  $\bar{B}$ , OUE  $\bar{k}$  在子信道  $\bar{n}$  上的信道容量可表示为

$$\bar{r}_{\bar{k}, \bar{n}} = \bar{B} \log \left( 1 + \frac{\bar{p}_{\bar{k}, \bar{n}} \bar{g}_{\bar{k}, \bar{n}}}{\bar{\sigma}_{\bar{n}}^2} \right), \forall \bar{k}, \bar{n} \quad (27)$$

其中,  $\bar{p}_{\bar{k}, \bar{n}}$ ,  $\bar{g}_{\bar{k}, \bar{n}}$ ,  $\forall \bar{k}, \bar{n}$  分别表示 OUE  $\bar{k}$  在子信道  $\bar{n}$  上的传输功率和信道增益,  $\bar{\sigma}_{\bar{n}}^2$  为子信道  $\bar{n}$  上的高斯白噪声。

由于每个 OUE 可同时利用多条子信道完成卸载, OUE  $\bar{k}$  总的传输速率可表示为

$$\bar{r}_{\bar{k}}(\bar{X}) = \sum_{\bar{n}=1}^{\bar{N}} \bar{x}_{\bar{k}, \bar{n}} \bar{r}_{\bar{k}, \bar{n}}, \forall \bar{k} \quad (28)$$

其中,  $\bar{x}_{\bar{k}, \bar{n}}$  为 OUE  $\bar{k}$  与子信道  $\bar{n}$  之间的配对决策。

$\bar{x}_{\bar{k}, \bar{n}} = 1$  表示 OUE  $\bar{k}$  选择子信道  $\bar{n}$  进行卸载,  $\bar{x}_{\bar{k}, \bar{n}} = 0$  则表示 OUE  $\bar{k}$  不选择该子信道。将  $\bar{t}_{\bar{k}}^{\text{ur}}$  和  $\bar{t}_{\bar{k}}^{\text{dr}}$  分别记为 OUE  $\bar{k}$  的上行链路传输时延和下行链路传输时延, 则有

$$\bar{t}_{\bar{k}}^{\text{ur}} = \frac{\bar{D}_{\bar{k}} \mu_{\bar{k}}}{\bar{r}_{\bar{k}}}, \forall \bar{k} \quad (29)$$

$$\bar{t}_{\bar{k}}^{\text{dr}} = \frac{\bar{D}_{\bar{k}} \mu_{\bar{k}} \beta_{\bar{k}}}{\bar{r}_{\bar{k}}}, \forall \bar{k} \quad (30)$$

其中,  $\beta_{\bar{k}}$  为 OUE  $\bar{k}$  处理 1 bit 原始数据后返回的数据量。OUE  $\bar{k}$  需要卸载的数据在 OECN 处理时延为

$$\bar{t}_{\bar{k}}^{\text{m}} = \frac{\bar{D}_{\bar{k}} \bar{a}_{\bar{k}} \mu_{\bar{k}}}{\bar{f}_{\text{m}}}, \forall \bar{k} \quad (31)$$

其中,  $\bar{f}_{\text{m}}$  为 OECN 计算速率。分别定义  $\bar{t}_{\bar{k}}^{\text{o}}$  和  $\bar{E}_{\bar{k}}^{\text{o}}$  为

OUE  $\bar{k}$  卸载消耗的总时延和总能耗（只考虑上行链路传输能耗），则有

$$\bar{t}_k^o = \bar{t}_k^{ur} + \bar{t}_k^{dr} + \bar{t}_k^m, \forall \bar{k} \quad (32)$$

$$\bar{E}_k^o = \bar{t}_k^{ur} \sum_{\bar{n}=1}^{\bar{N}} \bar{x}_{\bar{k},\bar{n}} \bar{p}_{\bar{k},\bar{n}}, \forall \bar{k} \quad (33)$$

综上，OUE  $\bar{k}$  执行总时延为本地处理部分总时延与卸载部分总时延中的最大值，记为

$$\bar{t}_k = \max(\bar{t}_k^1, \bar{t}_k^o), \forall \bar{k} \quad (34)$$

显然，当  $\bar{t}_k^o = \bar{t}_k^1$  时，两个并行过程花费相同的时延，OUE  $\bar{k}$  可达最小时延。求此时 OUE  $\bar{k}$  的卸载比  $\mu_{\bar{k}}$  及时延  $\bar{t}_k$ ，经过化简后可分别表示为

$$\mu_{\bar{k}} = \frac{\bar{r}_{\bar{k}}(\bar{X}) \bar{a}_{\bar{k}} \bar{f}_m}{\bar{f}_{\bar{k}} \bar{f}_m (1 + \beta_{\bar{k}}) + \bar{r}_{\bar{k}}(\bar{X}) \bar{a}_{\bar{k}} (\bar{f}_m + \bar{f}_{\bar{k}})}, \forall \bar{k} \quad (35)$$

$$\bar{t}_k = \frac{\bar{D}_{\bar{k}} \bar{a}_{\bar{k}} [(1 + \beta_{\bar{k}}) \bar{f}_m + \bar{r}_{\bar{k}}(\bar{X}) \bar{a}_{\bar{k}}]}{(1 + \beta_{\bar{k}}) \bar{f}_m \bar{f}_{\bar{k}} + \bar{r}_{\bar{k}}(\bar{X}) \bar{a}_{\bar{k}} (\bar{f}_m + \bar{f}_{\bar{k}})}, \forall \bar{k} \quad (36)$$

由式(35)、式(36)可知，通过求得最优本地计算速率  $\bar{f}_{\bar{k}}^*$  和最优传输速率  $\bar{r}_{\bar{k}}^*$  中的最优卸载决策，即可得出最优卸载比  $\mu_{\bar{k}}^*$  和最优时延  $\bar{t}_k^*$ 。

为了降低系统的时延消耗，保证 OUE 之间的公平性，将最小化每个 OUE 的最大时延作为优化目标函数。同时，为了便于计算，可定义

$$\bar{T}_{\bar{k}}(\bar{f}\bar{1}, \mu, \bar{X}) = \frac{1}{\bar{t}_k} = \frac{\bar{f}_{\bar{k}}}{\bar{D}_{\bar{k}} \bar{a}_{\bar{k}}} + \frac{\bar{r}_{\bar{k}}(\bar{X}) \bar{f}_m}{\bar{D}_{\bar{k}} [(1 + \beta_{\bar{k}}) \bar{f}_m + \bar{r}_{\bar{k}}(\bar{X}) \bar{a}_{\bar{k}}]}, \forall \bar{k} \quad (37)$$

基于上述讨论，联合本地计算速率  $\bar{f}\bar{1}$ 、卸载比  $\mu$  和 OUE 与子信道之间的配对决策  $\bar{X}$  的优化问题可表示为

$$\begin{aligned} & \max_{\{\bar{f}\bar{1}, \mu, \bar{X}\}} \min_{\bar{k}} (\bar{T}_{\bar{k}}(\bar{f}\bar{1}, \mu, \bar{X})), \forall \bar{k} \\ & \text{s.t. C1: } \bar{x}_{\bar{k},\bar{n}} \in \{0, 1\}, \forall \bar{k}, \bar{n} \\ & \text{C2: } \sum_{\bar{n}=1}^{\bar{N}} \bar{x}_{\bar{k},\bar{n}} \leq M, \forall \bar{k} \\ & \text{C3: } \sum_{\bar{k}=1}^{\bar{K}} \bar{x}_{\bar{k},\bar{n}} \leq 1, \forall \bar{n} \\ & \text{C4: } \bar{E}_{\bar{k}}^1 + \bar{E}_{\bar{k}}^o \leq \bar{E}_{\bar{k}}^{\max}, \forall \bar{k} \\ & \text{C5: } 0 \leq \bar{f}\bar{1}_{\bar{k}} \leq \bar{f}\bar{1}_{\bar{k}}^{\max}, \forall \bar{k} \\ & \text{C6: } 0 \leq \mu_{\bar{k}} \leq 1, \forall \bar{k} \\ & \text{C7: } \bar{t}_k \leq \tau_{\bar{k}}, \forall \bar{k} \end{aligned} \quad (38)$$

约束 C1 表明 OUE 与子信道之间的配对决策为二进制变量；约束 C2、C3 分别表示一条子信道最多只能被一个 OUE 选择，而每个 OUE 最多可同时选择  $M$  条子信道；约束 C4 表明每个 OUE 实际能耗不超过其最大能耗约束；约束 C5 表示每个 OUE 处理数据时的本地计算速率不超过其最大本地计算速率；约束 C6 为 OUE 卸载比例的取值范围；约束 C7 保证每个 OUE 处理数据的总时延不超过远海场景下网络连通时间。

## 5.2 远海 MSU 算法描述

针对同步优化二进制变量和连续变量的非线性规划问题，文献[22]提出了一种启发式算法。首先将二进制变量放宽为连续变量，然后利用拉格朗日对偶分解法解决该非线性规划问题。但该文献中的子信道分配并不合理且难以实现，因此，为了避免上述算法的不合理性，同时降低算法复杂度，本文将式(38)的优化问题划分为两个子问题进行解决，即最优本地计算速率分配问题与最优子信道分配问题。

### 5.2.1 本地计算速率分配优化

由式(37)可知，目标函数  $\bar{T}_{\bar{k}}(\bar{f}\bar{1}, \mu, \bar{X})$  随着  $\bar{f}\bar{1}$  增大而单调递增。此外，根据式(26)，本地处理能耗也随着  $\bar{f}\bar{1}$  的增大而增大。综上所述，当 OUE  $\bar{k}$  所允许的最大能耗全部用于本地处理数据（即  $\mu_{\bar{k}} = 0$ ）且满足式(38)约束 C5 时，可以得到最优的本地计算速率  $\bar{f}\bar{1}_{\bar{k}}^*$  为

$$\bar{f}\bar{1}_{\bar{k}}^* = \min \left( \sqrt{\frac{\bar{E}_{\bar{k}}^{\max}}{\bar{D}_{\bar{k}} \bar{a}_{\bar{k}} \bar{t}}}, \bar{f}\bar{1}_{\bar{k}}^{\max} \right), \forall \bar{k} \quad (39)$$

### 5.2.2 卸载决策与卸载比联合优化

远海场景下，节点资源匮乏、网络连通时间有限，规定远海场景下每个 OUE 可同时分配多条子信道，但 OUE 分配的子信道总数不超过  $M$  条。同时，为了保障各 OUE 之间的公平性，优先为通信条件较差的 OUE 分配最好的子信道卸载（只考虑信道增益  $\bar{g}_{\bar{k},\bar{n}}$ ），有利于缩短整体时延。该策略的主要思想如下：首先根据式(37)求得此时各 OUE 的优化目标值（在初始化的情况下，不为任何 OUE 分配信道资源，只比较各 OUE 本地处理时延，随着迭代次数的增加，不断更新各 OUE 与子信道之间的配对决策，该值也随之不断更新），然后通过比较选择数值最小的 OUE 分配信道最好的子信道，

同时保证 OUE 之间公平性。基于上述思想, 针对远海网络连通时间充足和有限的情况, 分别制定相应的子信道分配算法。

当网络连通时间  $\tau$  充足时, 在每次迭代过程中, 选出情况最恶劣的 OUE  $\bar{k}$ 。在满足该 OUE  $\bar{k}$  能耗约束与最大子信道分配数量约束条件的情况下, 令  $\bar{x}_{\bar{k}, \bar{n}} = 1$ , 此时计算 OUE  $\bar{k}$  的时延。若该 OUE 根据式(37)得出的目标函数值升高, 则将子信道分配给该 OUE 并进行数据更新 (包含当前目标函数值、OUE  $\bar{k}$  的子信道集  $\omega_{\bar{k}}$ 、剩余子信道集合  $\theta$  和信道容量等)。反之, 若 OUE  $\bar{k}$  不能满足约束条件或目标函数值未发生明显变化, 则表明 OUE  $\bar{k}$  已得到最优的子信道分配决策, 此时令  $\bar{x}_{\bar{k}, \bar{n}} = 0$ 。结束当前迭代并继续寻找剩余 OUE 的最优子信道分配决策, 直至所有 OUE 均获得最优的子信道分配决策。最终, 根据式(35)和式(36)计算每个 OUE 的最优卸载比和对应的时延, 并输出结果。具体算法描述如算法 5 所示。

**算法 5** 基于  $\tau$  充足时的子信道分配算法伪代码初始化

令  $\bar{x}_{\bar{k}, \bar{n}} = 0$ , 则  $\bar{r}_{\bar{k}} = 0$ , OUE  $\bar{k}$  分配子信道集  $\omega_{\bar{k}} = \emptyset, \forall \bar{k}, \bar{n}$   
 令未分配子信道集  $\theta = \{1, 2, \dots, \bar{N}\}$   
 根据式(37)计算各 OUE 的初始目标函数值  $\bar{T}_{\bar{k}} = \frac{\bar{\Pi}_{\bar{k}}^*}{\bar{D}_{\bar{k}} \bar{a}_{\bar{k}}}, \forall \bar{k}$   
 While( $\theta \neq \emptyset$ )  
 找到  $\bar{T}_{\bar{k}}$  最小值所对应的 OUE  $\bar{k}^*$   
 找到信道状况最好的子信道  $\bar{n}^*$   
 根据式(37)计算  $\bar{T}_{\text{new}} = \frac{\bar{\Pi}_{\bar{k}^*}^*}{\bar{D}_{\bar{k}^*} \bar{a}_{\bar{k}^*}} + \frac{\left( \bar{r}_{\bar{k}^*} + \bar{B} \text{lb} \left( 1 + \frac{\bar{P}_{\bar{k}^*, \bar{n}^*} \bar{g}_{\bar{k}^*, \bar{n}^*}}{\bar{N} \bar{\sigma}_{\bar{n}^*}^2} \right) \right) \bar{f}_m}{\bar{D}_{\bar{k}^*} \left[ \left( 1 + \beta_{\bar{k}^*} \right) \bar{f}_m + \left( \bar{r}_{\bar{k}^*} + \bar{B} \text{lb} \left( 1 + \frac{\bar{P}_{\bar{k}^*, \bar{n}^*} \bar{g}_{\bar{k}^*, \bar{n}^*}}{\bar{N} \bar{\sigma}_{\bar{n}^*}^2} \right) \right) \bar{a}_{\bar{k}^*} \right]}$   
 根据式(26)、式(33)、式(35)、式(39)计算  $\bar{E}_{\bar{k}^*}^l$  和  $\bar{E}_{\bar{k}^*}^o$   
 if  $\bar{E}_{\bar{k}^*}^l + \bar{E}_{\bar{k}^*}^o \leq \bar{E}_{\bar{k}^*}^{\text{max}}$  &  $\sum_{\bar{n}=1}^{\bar{N}} \bar{x}_{\bar{k}^*, \bar{n}} \leq M$  &  $\bar{T}_{\text{new}} > \bar{T}_{\bar{k}^*}$   
 更新信息:  $\theta = \theta - \{\bar{n}^*\}, \omega_{\bar{k}^*} = \omega_{\bar{k}^*} \cup \{\bar{n}^*\}, \bar{x}_{\bar{k}^*, \bar{n}^*} = 1$

$$\text{令 } \bar{r}_{\bar{k}^*} = \bar{r}_{\bar{k}^*} + \bar{B} \text{lb} \left( 1 + \frac{\bar{P}_{\bar{k}^*, \bar{n}^*} \bar{g}_{\bar{k}^*, \bar{n}^*}}{\bar{N} \bar{\sigma}_{\bar{n}^*}^2} \right),$$

$$\bar{T}_{\bar{k}^*} = \bar{T}_{\text{new}}$$

else

continue

end

$$\mu_{\bar{k}^*} = \frac{\bar{r}_{\bar{k}^*} \bar{a}_{\bar{k}^*} \bar{f}_m}{\bar{\Pi}_{\bar{k}^*}^* \bar{f}_m (1 + \beta_{\bar{k}^*}) + \bar{r}_{\bar{k}^*} \bar{a}_{\bar{k}^*} (\bar{f}_m + \bar{\Pi}_{\bar{k}^*}^*)}$$

end

当网络连通时间  $\tau$  有限时, 需对上述算法进行相应的调整。首先, 判断当前情况最恶劣的 OUE 是否满足网络连通时间约束。若满足, 则执行网络连通时间充足情况下的子信道分配算法。反之, 为降低计算复杂度, 将所有子信道根据信道状况递减排序。在满足能耗约束和最大子信道分配数量约束的条件下, 为该 OUE 分配子信道以降低数据传输时延, 直至该 OUE 满足网络连通时间约束。将 OUE 所分配的子信道对应的卸载决策均记为 1, 跳出本次迭代并对相应的参数进行更新。若该 OUE 利用最大可分配数量的子信道卸载仍不能满足网络连通时间约束, 则将 OUE 的数据进行本地处理, 即  $\bar{x}_{\bar{k}, \bar{n}} = 0, \forall \bar{n}$ , 不为其分配任何信道资源, 并将该部分信道资源用于剩余 OUE 卸载。之后, 继续按照相同的步骤依次为其余待处理的 OUE 分配剩余的子信道, 直至所有 OUE 完成最优子信道分配。此时, 输出每个 OUE 的最优卸载比及对应的时延。算法具体伪代码如算法 6 所示。

**算法 6** 基于  $\tau$  有限时的子信道分配算法伪代码初始化

令  $\bar{x}_{\bar{k}, \bar{n}} = 0$ , 则  $\bar{r}_{\bar{k}} = 0$ , OUE  $\bar{k}$  分配子信道集  $\omega_{\bar{k}} = \emptyset, \forall \bar{k}, \bar{n}$   
 令未分配子信道集  $\theta = \{1, 2, \dots, \bar{N}\}$   
 根据式(37)计算各 OUE 的初始目标函数值  $\bar{T}_{\bar{k}} = \frac{\bar{\Pi}_{\bar{k}}^*}{\bar{D}_{\bar{k}} \bar{a}_{\bar{k}}}, \forall \bar{k}$   
 While( $\theta \neq \emptyset$ )  
 找到  $\bar{T}_{\bar{k}}$  最小值所对应的 OUE  $\bar{k}^*$   
 if  $\bar{T}_{\bar{k}^*} \geq 1 / \tau_{\bar{k}^*}$   
 执行网络连通时间充足下的子信道分配算法

else

将OUE $\bar{k}^*$ 所对应子信道以信道状况降序排列

if OUE $\bar{k}^*$ 找到满足网络联通时间、能耗、最大允许分配子信道数约束条件下的最小子信道分配组群 $\mathbf{v}_{\bar{k}^*}$

更新信息:  $\boldsymbol{\theta} = \boldsymbol{\theta} - \{\mathbf{v}_{\bar{k}^*}\}, \bar{x}_{\bar{k}^*, \bar{n}^*} = 1,$

$$\bar{n}^* \in \mathbf{v}_{\bar{k}^*}, \boldsymbol{\omega}_{\bar{k}^*} = \boldsymbol{\omega}_{\bar{k}^*} \cup \mathbf{v}_{\bar{k}^*}$$

$$\bar{r}_{\bar{k}^*} = \sum_{\bar{n} \in \mathbf{v}_{\bar{k}^*}} \bar{x}_{\bar{k}^*, \bar{n}} \bar{B} \text{lb} \left( 1 + \frac{\bar{p}_{\bar{k}^*, \bar{n}} \bar{g}_{\bar{k}^*, \bar{n}^*}}{\bar{N} \bar{\sigma}_{\bar{n}^*}^2} \right),$$

$$\bar{T}_{\bar{k}^*} = \frac{\bar{f}_{\bar{k}^*}^*}{\bar{D}_{\bar{k}^*} \bar{a}_{\bar{k}^*}} + \frac{\bar{r}_{\bar{k}^*} \bar{f}_m}{\bar{D}_{\bar{k}^*} [(1 + \beta_{\bar{k}^*}) \bar{f}_m + \bar{r}_{\bar{k}^*} \bar{a}_{\bar{k}^*}]},$$

else

OUE $\bar{k}^*$ 只能本地处理, 不分配任何信道资源

continue

end

$$\mu_{\bar{k}^*} = \frac{\bar{r}_{\bar{k}^*} \bar{a}_{\bar{k}^*} \bar{f}_m}{\bar{f}_{\bar{k}^*}^* \bar{f}_m (1 + \beta_{\bar{k}^*}) + \bar{r}_{\bar{k}^*} \bar{a}_{\bar{k}^*} (\bar{f}_m + \bar{f}_{\bar{k}^*}^*)}, \forall \bar{k}^*$$

end

end

## 6 实验仿真与结果分析

### 6.1 近海 MSU 仿真与结果分析

近海 MSU 仿真设置  $K = 5, S = 10, N = 100,$  即同时有 50 个子任务向 OECN 的 MAC 服务器发送卸载请求, 且有 100 条子信道供子任务卸载选择。近海 MSU 方案参数取值如表 2 所示。

表 2 近海 MSU 方案参数取值

参数	取值
平均计算密度 $\mathbf{a}$	[15, 25] cycles/bit
SBS 计算能力 $\mathbf{f}_c$	[20, 30] GHz
子任务最大能耗 $E_{\max}$	0.2 J
子任务额定功率 $\mathbf{p}_{\max}$	[0.5, 1] W
OUE 计算能力 $\mathbf{f}$	[0.05, 0.1] GHz
子信道带宽 $\mathbf{B}$	[1, 2] MHz
信道增益噪声比 $\frac{g}{\sigma^2}$	[10, 200] W <sup>-1</sup>

设 MO-AFSA 中人工鱼数量 fishNum = 30, 人工鱼的视野范围 visual = 50, 步长 step  $\in [1, 7]$ , 拥挤度  $\delta = 2.62$ , 最大尝试次数 tryNum = 100, 最大迭代次数 MAX\_GEN = 100, 惩罚函数中惩罚因子 g1 = 100, 变异概率 pm 分别为 0.000 02、0.000 06、0.006、0.000 02、0.004。另外, 子任务数据量  $\mathbf{D}$  (单位: KB) 分别为 [300,500]、[500,700]、[700,900]、[900,1 100]、[1 100,1 300]。为验证 MO-AFSA 在近海卸载方案中的有效性, 将该算法与粒子群优化 (PSO, particle swarm optimization) 算法<sup>[23]</sup>、AFSA<sup>[19]</sup>、改进步长的人工鱼群算法 (ISS-AFSA, artificial fish swarm algorithm with improved step size)、无优先级的基于变异操作的人工鱼群算法 (NP-MO-AFSA, mutation-artificial fish swarm algorithm with no priority) 进行性能比较。

由式(18)可知, 该模型的适应度函数为: 每个子任务在惩罚因子约束下, 本地处理时延与卸载总时延的差值, 并以每个 OUE 为单位求和取平均。适应度函数数值越大, 表明该 OUE 节省时延效果越明显。同时, 为了保证各 OUE 之间的公平性, 本节将多 OUE 的适应度函数求和取平均, 分析其整体节省时延效果和迭代收敛效果。

不同算法及子任务数据量下的多 OUE 平均适应度函数值对比如图 2 所示。在信道资源充足的情况下, 多 OUE 平均适应度函数值随子任务数据量增大而增大。由此可知, 当 OUE 本地计算能力不足而卸载节点计算资源充足时, 子任务数据量越大, 卸载节省时延效果更加显著。图 2 对比了多种算法, 可以发现, AFSA 和 PSO 算法由于各参数相对固定, 跳出局部最优解的能力受限。ISS-AFSA 虽然通过随机步长改善了跳出局部最优的能力, 但效果一般。MO-AFSA 令步长随机化且引入遗传算法中的变异操作, 跳出局部最优的能力进一步增强, 从而可获得最优适应度函数值。此外, 为进一步提升 OUE 体验品质, 该算法通过设置优先级函数使时延敏感型 OUE 可优先选择有利于其卸载的子信道, 相较 NP-MO-AFSA, 收益效果显著。

由于 OUE 本地计算能力不足, 各算法下的适应度函数值差异不明显 (差异主要体现在卸载时延差异)。多 OUE 多算法迭代性能对比如图 3 所示, 进一步分析了当  $K = 4, S = 10, \mathbf{D} = [500,700]$  时, MO-AFSA 的迭代收敛性及节省时延效果。可以看

出，相比其他算法，MO-AFSA 收敛速度最快，迭代 23 次后平均适应度函数值便达到最优。此外，经过多次迭代后，MO-AFSA 的多 OUE 平均适应度函数值也明显最大，进一步说明 MO-AFSA 节省时延效果最好。其他算法由于容易陷入局部最优，收敛效果和最优适应度函数值较 MO-AFSA 均存在较大差距。

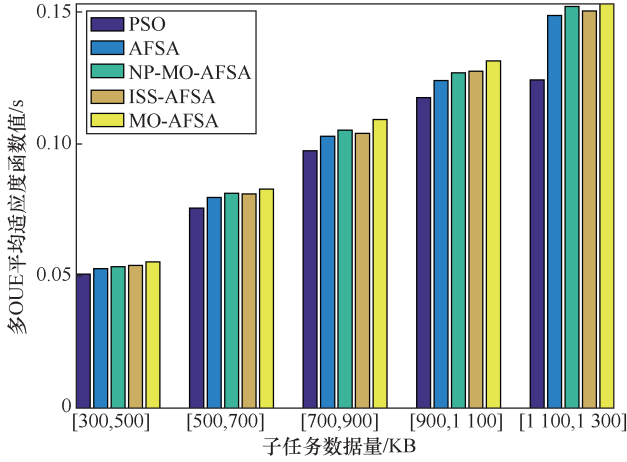


图 2 不同算法及子任务数据量下的多 OUE 平均适应度函数值对比

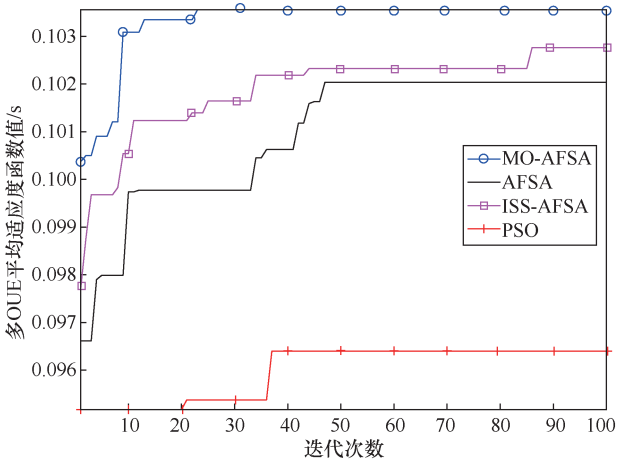


图 3 多 OUE 多算法迭代性能对比

### 6.2 近海 MMU 仿真与结果分析

在仿真实验中，设置  $\tilde{K}=4$ ，且每个 OUE  $\tilde{k}$  划分子信道总数  $\tilde{N}_{\tilde{k}}$  均为 100，每个 OUE 同时有多个子任务向 OECN 的 MAC 服务器发送卸载请求，且有 100 条子信道供子任务卸载选择。近海 MMU 方案参数取值如表 3 所示。

设 MO-AFSA 中人工鱼数量 fishNum1=30，人工鱼的视野范围 visual1=50，步长 step1 ∈ [1, 9]，拥挤度 δ1=0.382，最大尝试次数 tryNum1=50，最大迭代次数 MAX\_GEN1=100，惩罚函数中惩罚

因子  $\tilde{g}1=100$ ，变异概率 pm1 分别为 0.000 2、0.001、0.000 1、0.006、0.000 2，子任务数  $\tilde{S}$  分别为 5、10、15、20、25。

表 3 近海 MMU 方案参数取值

参数	取值
平均计算密度 $\tilde{a}$	[5, 20] cycles/bit
SBS 计算能力 $\tilde{f}_c$	50 GHz
子任务最大能耗 $\tilde{E}_{max}$	0.2 J
子任务额定功率 $\tilde{p}_{max}$	[0.5, 1] W
移动节点计算能力 $\tilde{f}_m$	[1, 15] GHz
OUE 计算能力 $\tilde{f}$	[0.01, 0.1] GHz
子信道带宽 $\tilde{B}$	[1, 2] MHz
信道增益噪声比 $\frac{\tilde{g}}{\sigma^2}$	[10, 200] W <sup>-1</sup>
任务数据量 $\tilde{D}$	[500, 700] KB
移动节点到固定节点的传输时延 tts	0.008 s

多 OUE 在不同  $\tilde{S}$  及不同算法下的平均时延对比如图 4 所示。在信道资源充足的情况下，随着子任务划分数量增多，各任务不断争夺优异信道状况的子信道，并且在卸载节点排队等候的概率和时延增大，导致平均时延随之增大。MO-AFSA 具有较强的寻优能力，通过为各 OUE 设置优先级解决计算资源差异明显问题，同时令各子任务依据先到先得策略在一级 OECN 的 MAC 服务器排队等候处理。因此，平均时延明显低于其他算法。

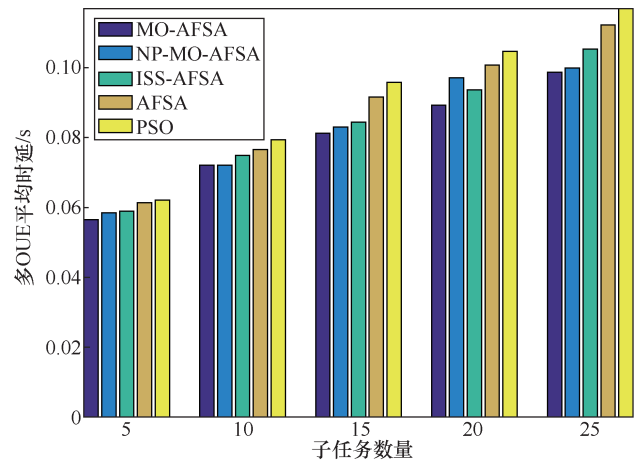


图 4 多 OUE 在不同  $\tilde{S}$  及不同算法下的平均时延对比

多 OUE 在不同  $\tilde{S}$  及不同卸载策略下的平均时延对比如图 5 所示。对比策略包括：1) 随机卸载

(RANDOM, random) 策略——每个 OUE 的子任务随机分配至本地、移动节点以及固定节点，并随机分配一个子信道传输；2) 基站卸载(BSprocess, base station process) 策略——子任务将移动节点作为中继，传输至 SBS 进行处理，采用 MO-AFSA 分配子信道；3) 移动节点卸载(MNprocess, mobile node process) 策略——子任务将移动节点作为计算节点，采用 MO-AFSA 分配子信道；4) 本地执行 (LOCAL, local) 策略——子任务全部本地执行，不进行计算卸载。

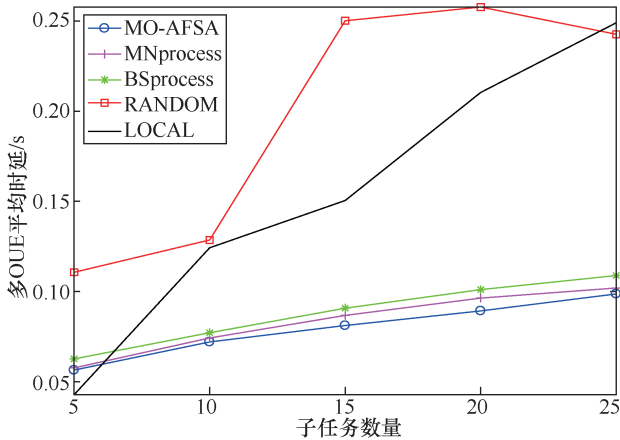


图 5 多 OUE 在不同  $\bar{s}$  及不同卸载策略下的平均时延对比

如图 5 所示，5 种策略下的平均时延均随着子任务数量的增加而增大。由于本地计算能力受限，为便于比较，图 5 呈现的 LOCAL 策略结果为缩小至 1/15 后的效果，相较卸载处理策略，时延最大。RANDOM 策略具有随机性，不同子任务数量下的时延存在波动。移动节点与 SBS 之间的传输时延较大，虽然 SBS 计算资源充足，但仅利用 BSprocess 卸载策略不能获得最佳收益。移动节点计算资源不足，各子任务在 MAC 服务器上存在排队等候处理时延，该时延随着子任务数量增多而增大，仅利用 MNprocess 策略同样不能获得最佳收益。MO-AFSA 结合上述两种卸载策略的优势，并通过设置优先级和令子任务依据先到先得策略在 MAC 服务器上处理数据，弥补 MNprocess 策略的缺陷，获得最佳收益。

### 6.3 远海 MSU 仿真与结果分析

仿真实验设置  $\bar{N} = 100$ ，即同时有多个 OUE 向 OECN 的 MAC 服务器发出卸载请求，且有 100 条子信道供选择，远海 MSU 方案参数取值如表 4 所示。

表 4 远海 MSU 方案参数取值

参数	取值
平均计算密度 $\bar{a}$	[5, 15] cycles/bit
芯片结构参数 $\bar{l}$	$10^{-26}$
OUE 最大能耗 $\bar{E}_{\max}$	0.2 J
传输功率 $\bar{p}$	$[0.5/\bar{N}, 1/\bar{N}]$ W
移动节点计算能力 $\bar{f}_c$	[5, 8] GHz
本地最大计算速率 $\bar{f}_{\max}$	[0.01, 0.1] GHz
子信道带宽 $\bar{B}$	1 MHz
信道增益噪声比 $\frac{\bar{g}}{\sigma^2}$	[10, 200] $W^{-1}$
任务数据量 $\bar{D}$	[500, 700] KB
处理 1 bit 数据返回数据比 $\beta$	0.3

在  $\tau$  充足情况下，不同  $\bar{K}$  及  $M$  条件下多 OUE 平均时延对比如图 6 所示。当  $\bar{K} = 1$  时，同时为 OUE 选择信道状况最优的  $M$  条子信道传输数据，节省时延效果最佳。随着 OUE 数量增多，卸载需要的信道资源增多，只能不断选择信道状况次之的子信道卸载，多个 OUE 平均时延不断增大。在信道资源充足的情况下， $M$  越大，每个 OUE 可同时将数据分配给更多子信道传输，多个 OUE 平均时延不断降低。值得注意的是，当  $M=3$  时， $\bar{K} = 25$  信道资源相较  $\bar{K} = 35$  情况充足，但  $\bar{K} = 35$  时信道资源的利用率更高，更多 OUE 可分配子信道卸载，缩短时延效果更佳。

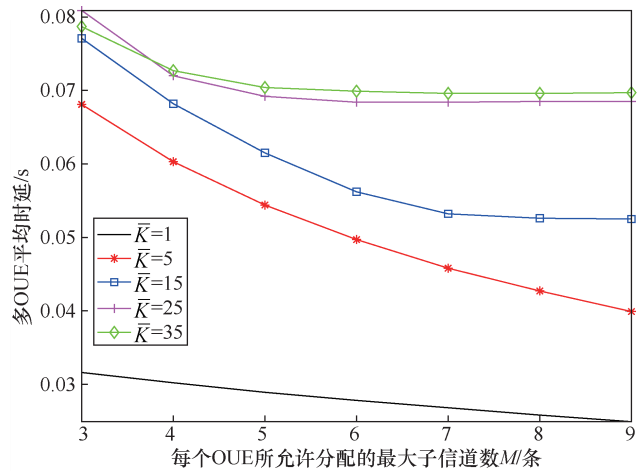


图 6 不同  $\bar{K}$  及  $M$  条件下多 OUE 平均时延对比

以  $M = 6$ ， $\bar{K} = 15$ ， $\bar{N} = 50$  为例，不同  $\tau$  条件下各 OUE 时延对比如图 7 所示。 $\tau = 0.07$  时信道资源相对充足，各 OUE 可通过同时分配多条子信道，在满足网络连通时间约束和能耗约束下完成卸载，

缩短时延。随着 $\tau$ 降低,各 OUE 即使同时利用 $M$ 条子信道卸载也不能满足网络连通时间约束,无法卸载的 OUE 不断增多。此外,当 $\tau$ 较小时,不可卸载的 OUE 增多,则可卸载的 OUE 可分配的信道资源增多。因此,该部分 OUE 节省时延效果增强,如 $\tau=0.05$ 时。该算法通过对信道状况递减排序,使 OUE 能够尽快找到最优子信道卸载,当 OUE 不能卸载处理时,预先分配给该 OUE 的信道资源将保留,用于解决下一个 OUE 的卸载问题。

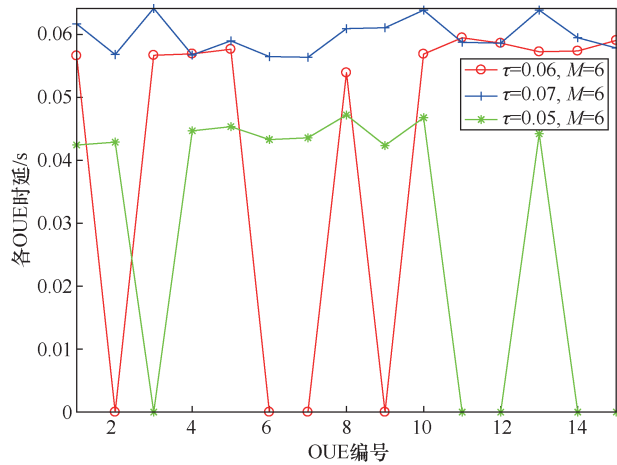


图7 不同 $\tau$ 条件下各 OUE 时延对比

## 7 结束语

本文分别建立了近海、远海场景下基于 OFDMA 技术的多用户卸载模型。在近海场景下,针对计算资源充足与有限两种情况,分别提出 MSU 和 MMU 卸载方案。将优化问题划分为两个子问题,通过数学优化解决传输功率优化问题;通过改进传统人工鱼群算法,提出 MO-AFSA 解决卸载决策分配问题。在远海场景下,分别提出针对网络连通时间充足与有限情况下的子信道分配算法,解决远海 MSU 卸载决策分配问题。仿真结果表明,近海所提算法对比其他算法,所提策略对比其他策略,节省时延效果明显增强;远海仿真结果分析了不同限制因素对时延消耗的影响,对实际应用具有重要意义。未来研究方向主要针对远海场景节点资源相对丰富的情况,拟考虑解决多 OUE 与多个中大型船舶之间的卸载问题,也将进一步分析海洋环境因素对卸载的影响,拟考虑融合通信资源分配因素的卸载,为更好地提升我国海洋观测能力奠定坚实的基础。

## 参考文献:

- [1] PORAMBAGE P, OKWUIBE J, LIYANAGE M, et al. Survey on multi-access edge computing for Internet of things realization[J]. IEEE Communications Surveys and Tutorials, 2018, 20(4): 2961-2991.
- [2] TALEB T, SAMDANIS K, MADA B, et al. On multi-access edge computing: a survey of the emerging 5G network edge cloud architecture and orchestration[J]. IEEE Communications Surveys and Tutorials, 2017, 19(3): 1657-1681.
- [3] SAFAVAT S, SAPAVATH N N, RAWAT D B, et al. Recent advances in mobile edge computing and content caching[J]. Digital Communications Networks, 2020, 6(2): 189-194.
- [4] HUANG L, FENG X, ZHANG C, et al. Deep reinforcement learning-based joint task offloading and bandwidth allocation for multi-user mobile edge computing[J]. Digital Communications Networks, 2019, 5(1): 10-17.
- [5] ABBAS N, ZHANG Y, TAHERKORDI A, et al. Mobile edge computing: a survey[J]. IEEE Internet of Things Journal, 2017, 5(1): 450-465.
- [6] ZHOU Y, LIU L, WANG L, et al. Service aware 6G: an intelligent and open network based on convergence of communication, computing and caching[J]. Digital Communications Networks, 2020, 6(3): 253-260.
- [7] KEWEI S, ANDREW Y T, WEI W, et al. A survey of edge computing-based designs for IoT security[J]. Digital Communications Networks, 2020, 6(2): 195-202.
- [8] KAMOUN M, LABIDI W, SARKISS M. Joint resource allocation and offloading strategies in cloud enabled cellular networks[C]// Proceedings of IEEE International Conference on Communications (ICC). Piscataway: IEEE Press, 2018.
- [9] WANG J Y, PENG J, WEI Y H, et al. Adaptive application offloading decision and transmission scheduling for mobile cloud computing[J]. China Communications, 2017, 14(3): 169-181.
- [10] CAO S W, TAO X F, HOU Y Z, et al. An energy-optimal offloading algorithm of mobile computing based on HetNets[C]// Proceedings of 2015 International Conference on Connected Vehicles and Expo (ICCVE). Piscataway: IEEE Press, 2015.
- [11] MAO Y Y, ZHANG J, LETAIEF K B. Joint task offloading scheduling and transmit power allocation for mobile-edge computing systems[C]// Proceedings of 2017 IEEE Wireless Communications and Networking Conference (WCNC). Piscataway: IEEE Press, 2017.
- [12] WANG Y T, SHENG M, WANG X J, et al. Mobile-edge computing: partial computation offloading using dynamic voltage scaling[J]. IEEE Transactions on Communications, 2016, 64(10): 4268-4282.
- [13] MAO Y Y, ZHANG J, SONG S H, et al. Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems[J]. IEEE Transactions on Wireless Communications, 2017, 16(9): 5994-6009.
- [14] REN J K, YU G D, CAI Y L, et al. Latency optimization for resource allocation in mobile-edge computation offloading[J]. IEEE Transactions on Wireless Communications, 2018, 17(8): 5506-5519.

- [15] LI Q P, ZHAO J H, GONG Y, et al. Energy-efficient computation offloading and resource allocation in fog computing for Internet of everything[J]. China Communications, 2019, 16(3): 32-41.
- [16] YOU C S, HUANG K B, CHAE H, et al. Energy-efficient resource allocation for mobile-edge computation offloading[J]. IEEE Transactions on Wireless Communications, 2017, 16(3): 1397-1411.
- [17] HE W, ZHANG Y Z, HUANG Y H, et al. latency minimization for full-duplex mobile-edge computing system[C]//Proceedings of 2019 IEEE International Conference on Communications (ICC). Piscataway: IEEE Press, 2019.
- [18] PAN Y J, CHEN M, YANG J H, et al. Energy-efficient NOMA-based mobile edge computing offloading[J]. IEEE Communications Letters, 2019, 23(2): 310-313.
- [19] 周晓敏. 面向节能的移动边缘计算的卸载策略研究[D]. 北京: 北京邮电大学, 2019.  
ZHOU X M. Research on offloading strategy in energy-saving mobile edge computing system[D]. Beijing: Beijing University of Posts and Telecommunications, 2019.
- [20] SU X, MENG L L, HUANG J, et al. Intelligent maritime networking with edge services and computing capability[J]. IEEE Transactions on Vehicular Technology, 2020, 69(11): 13606-13620.
- [21] DAI Y P, SHENG M, LIU J Y, et al. Resource allocation for low latency mobile edge computation offloading in NOMA networks[C]//Proceedings of 2018 IEEE Global Communications Conference (GLOBECOM). Piscataway: IEEE Press, 2018.
- [22] WANG Y T, SHENG M, WANG X J, et al. Cooperative dynamic voltage scaling and radio resource allocation for energy-efficient multiuser mobile edge computing[C]//Proceedings of 2018 IEEE International Conference on Communications (ICC). Piscataway: IEEE Press, 2018: 1938-1883.
- [23] 元晋, 孙海蓉, 巩锬, 等. 移动边缘计算中基于信誉值的智能计算卸载模型研究[J]. 通信学报, 2020, 41(7): 141-151.  
QI J, SUN H R, GONG K, et al. Research on intelligent offloading model based on reputation value in mobile edge computing[J]. Journal

on Communications, 2020, 41(7): 141-151.

#### [作者简介]



苏新 (1986-), 男, 博士, 河海大学副教授、硕士生导师, 主要研究方向为移动通信、边缘/雾计算、智慧海洋等。



王子怡 (1997-), 女, 河海大学硕士生, 主要研究方向为海洋网络、边缘/雾计算、计算卸载等。



王宇鹏 (1981-), 男, 博士, 沈阳航空航天大学教授, 辽宁省空天信息感知与智能处理重点实验室副主任, 主要研究方向为移动通信技术、物联网、自组织网络等。



周思源 (1985-), 男, 博士, 河海大学副教授, 河海大学无线通信与智能系统研究所副所长, 主要研究方向为车路协同自动驾驶、多天线通信技术、边缘计算等。