

面向海洋观监测传感网的计算卸载方法研究

苏新¹, 薛溟阳¹, 周一青², 朱金秀¹

(1. 河海大学物联网工程学院, 江苏 常州 231022; 2. 中国科学院计算技术研究所, 北京 100190)

摘要: 针对海洋网络节点间计算能力与通信资源的差异性, 提出了一种基于海洋网络连通概率的边缘计算节点选取方法。根据海洋近岸与远岸的网络节点密度不同, 分别建立 2 种卸载模型。在近岸场景下, 提出多节点协同的卸载方法, 利用基于海洋多节点协同卸载遗传算法求解; 在远岸场景下, 提出可容错的卸载方法, 利用基于分组交叉学习粒子群算法求解。仿真结果表明, 所提方法较传统方法可降低 30% 以上的网络延时并可节约 20% 以上的网络成本, 极大提高了各类海事应用的用户体验。

关键词: 海洋网络; 多接入边缘计算; 计算卸载; 遗传算法; 粒子群优化

中图分类号: TN929.52

文献标识码: A

DOI: 10.11959/j.issn.1000-436x.2021067

Research on computing offloading method for maritime observation monitoring sensor network

SU Xin¹, XUE Haoyang¹, ZHOU Yiqing², ZHU Jinxiu¹

1. College of IoT Engineering, Hohai University, Changzhou 213022, China

2. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China

Abstract: Considering the differences in computing capacity and communication resources of the maritime network nodes, a maritime network connectivity probability based method was proposed for selecting edge computing service nodes. Because of the different node densities in the near-shore and far-shore scenarios, two offloading models were established accordingly. In the near-shore scenario, a multi-node cooperative offloading method was proposed by using the genetic algorithm based on maritime multi-node cooperative offloading. In the far-shore scenario, a fault-tolerant offloading method was proposed based on the particle swarm algorithm with grouping cross learning. Simulation results show that compared with conventional methods, the proposed methods save over 30% network delay and reduces about 20% network costs, which can greatly enhance the maritime user experiences.

Keywords: maritime network, multi-access edge computing, computing offloading, genetic algorithm, particle swarm optimization

1 引言

海洋观监测传感网作为海洋信息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、紧急救援以及面向某些领域的低时延信息服务需求, 这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的要求。进入 5G/6G 时代^[1-2], 海洋观监测传感网络终端设备 (OUE, oceanic user equipment) 对于低时延高可靠的应用需求随之增加, 需要全新的数据处理方式以满足相关应用需求。

收稿日期: 2020-10-27; 修回日期: 2021-01-05

基金项目: 国家重点研发计划基金资助项目 (No.2021YFE0105500); 国家自然科学基金资助项目 (No.61801166)

Foundation Items: The National Key Research and Development Program of China (No.2021YFE0105500), The National Natural Science Foundation of China (No.61801166)

2014 年, 欧洲电信标准协会提出了移动边缘计算概念, 并于 2016 年扩展为多接入边缘计算 (MAEC, multi-access edge computing) [3]。其主要思想是将云计算的部分业务下沉至网络边缘, 在用户边缘分布式地部署服务与存储, 以快速地为提供服务, 极大减少回传网络带宽[4]。在网络边缘就近处理数据的边缘计算方式可实现终端用户对低时延、高可靠的要求[5]。因此, 将边缘计算关键技术引入海洋观监测传感网以满足低时延、高可靠的海事应用需求具有极高的研究意义和应用价值。

计算卸载技术作为边缘计算的关键技术, 对降低用户执行时延, 提升用户服务质量起着重要作用[6]。现阶段边缘计算卸载技术研究主要围绕陆地组网与车联网, 普遍根据其组网特征进行研究[7-8]。由于海洋环境复杂多变, 不能简单地将现有陆地组网中的 MAEC 卸载技术应用于海洋观监测传感网。因此, 将 MAEC 卸载机制引入海洋观监测传感网的研究会面临以下挑战。

1) 海洋场景节点密度变化明显, 边缘计算资源分布不均, 调度困难, 网络节点计算与通信资源存在差异。例如, 各个节点对时延和能耗的敏感度不同, 网络局部产生的大量数据会导致网络超负荷处理, 系统面临瘫痪危险。

2) 海洋观监测传感网终端设备移动速度不同, 网络拓扑变化快, 节点间通信范围存在差异且有效通信时间不同, 容易导致传输链路中断。

通过参考 MAEC 卸载技术在陆地组网及车联网中的研究[7], 结合海洋观监测传感网中存在的挑战, 本文以满足低时延、高可靠的海事应用需求为目标, 研究了海洋观监测传感网中的 MAEC 卸载技术, 主要贡献如下。

1) 针对节点密度差异化下的卸载问题, 提出了以下 2 种卸载模型: 在网络节点高密度的近岸区域, 计算资源丰富, 多用户计算卸载可用多个独立的单用户多边缘节点模型等效; 在网络节点低密度的远岸区域, 计算资源匮乏, 多用户计算卸载需考虑协同, 提出远岸双跳多用户模型。同时, 阐述了一种基于海洋网络连通概率的边缘节点选取方法, 使终端用户可选取该区域内有利于其卸载传输的最佳节点。

2) 提出了节点间有效通信时间约束, 并分别引入近岸和远岸 2 种卸载模型, 保证了用户数据卸载时网络的连通效率。

3) 在选取边缘计算节点基础上, 提出了 2 种计算卸载算法, 在 2 种卸载模型下实现了用户卸载时的最佳决策和资源调度, 解决了节点间资源差异导致的网络成本增加的问题。

2 相关工作

计算卸载相关研究工作按照系统模型可分为单用户设备 (SUE, single user equipment) 计算卸载和多用户设备 (MUE, multi-user equipment) 计算卸载, 本文将分别对这 2 种工作进行分析与讨论。

2.1 单用户设备计算卸载

目前, 针对 SUE 计算卸载的研究主要分为单用户与单个 MAEC 服务器 (简称为 SUE-SMAEC)、单用户与多个 MAEC 服务器 (简称为 SUE-MMAEC)。

文献[9]针对 SUE-SMAEC, 运用一维查询算法根据 UE 和 MAEC 服务器的计算资源及其之间的信道特征找到最优的卸载策略, 实现最小化执行时延的目标。然而, 文献[9]未考虑能耗约束会导致 UE 能耗过高。为同时满足 UE 对时延与能耗的要求, 文献[10]提出了在满足应用执行时延的同时最小化用户能耗的卸载策略。优化方法引入了在线学习和离线预先计算 2 种卸载方案。与全部卸载相比[9-10], 部分卸载更灵活, 允许用户将任务在本地和 MAEC 服务器分别执行。文献[11]假设 UE 应用由多个子任务组成, 各任务之间相互依赖, 并提出了基于二元粒子群的卸载算法, 使部分卸载相对于全部卸载具有显著的节能效果。

针对车联网下的计算卸载, 文献[4]研究了在 SUE-MMAEC 中任务分配和卸载决策的问题。文献[4]虽然考虑了节点移动性, 但并未对其可能产生卸载中断问题进行分析解决。针对车联网中车辆快速移动导致的卸载中断问题, 文献[12]提出了一种避免传输中断的自适应卸载策略, 考虑车速、传输速率等因素对卸载的影响, 并根据这些因素动态调整卸载策略, 避免卸载过程中的传输中断。

目前, SUE 计算卸载的研究主要围绕陆地组网和车联网 2 种场景, 且多数卸载方案仅考虑 SUE-SMAEC 的全部卸载和部分卸载, 较少考虑多 MAEC 服务器协同处理的卸载方式。

2.2 多用户设备计算卸载

通常, 由于 MAEC 系统中通信和计算资源有限, 难以满足过多终端用户的计算卸载需求, 且分布式计算场景之间难以达到 MAEC 资源的有效利

用^[13]。因此，学者们开始关注 MUE 场景下的卸载研究，以实现资源的合理分配。

文献[14]将 MUE 系统中资源分配表述为一个非线性问题，提出一种基于贪婪策略的资源分配方案，有效地降低了终端能耗。不同于通过贪婪策略进行资源分配，文献[15]提出了阈值结构的资源分配策略。在时分多址 MUE 系统中，通过本地计算能力和信道增益的卸载优先级函数确定用户优先级，基站根据用户优先级与阈值间的关系，决定用户卸载的数据量。文献[16]综合考虑卸载决策、信道状态、计算资源等影响因素，将超密集网络下的计算卸载模型表述为一个混合整数非线性规划问题，设计了一种结合遗传和粒子群的优化算法，以此实现最优资源分配。

以上方案均从降低系统时延或能耗角度出发，并未从用户对能耗和时延的敏感程度差异角度对系统时延和能耗进行综合优化。文献[17]在 MUE-SMAEC 系统中，首先针对用户对能耗和时延差异度，为其设置相应能耗时延优化比；再将卸载决策和资源分配的优化表述为非凸优化问题，通过鲸鱼遗传算法解决不同用户需求；最终实现资源分配。但文献[17]中算法的迭代速度相对较慢，增加了系统成本。针对计算资源分布不均的问题，文献[18]在 MUE-MMAEC 系统中通过主辅 MAEC 服务器协同处理的方式，运用模拟退火算法进行卸载决策和资源分配的联合优化，解决了资源分布不均导致的资源调度困难问题。

基于以上论述，现有 MUE 计算卸载研究主要以减少终端能耗和任务执行时延为优化目标，对综

合考虑优化任务时延和终端能耗的研究较少，且均未考虑 MUE 系统中计算卸载的可靠性。算法迭代速度都相对较慢，难以保证较小的计算卸载成本，无法满足海洋观测监测传感网的各类海事应用需求。

3 海洋网络系统模型

图 1 展示了本文提出的海洋网络系统模型，通常，距离海岸 37.04 km（即 20 海里）为近岸海域，海域界限向外一侧的为远岸海域^[19]。海洋网络中，不同海域节点密度差异明显，近岸海域由于海事活动较频繁，通常其网络节点密度高于远岸海域。

在海洋网络中，网络节点一般包含大型船舶、浮标、小型船只、水下机器人以及基站等。小型船只和浮标由于自身资源有限，一般难以独立完成数据处理，需要传输至其他节点进行处理。大型船舶与基站等相比浮标等节点计算能力较强，可充当边缘计算节点来处理终端用户卸载的数据。通常，基站计算能力强于大型船舶，在近岸海域，可通过多个海岸基站与大型船舶协同实现用户的计算卸载；在远岸海域，由于海域范围广，大规模建设成本投入大等因素，本文考虑在远岸海事活动频繁的海域建设 1~2 个大型海上基站保证海事应用需求，尽可能地利用现有设施满足海事应用需求。

结合相关工作讨论可知，单用户模型一般用于解决资源丰富情况下如何实现高效卸载决策与资源调度，以满足 OUE 应用需求的问题。近岸海域计算资源丰富，某一区域内可用计算资源较多，需要考虑 OUE 如何充分利用资源，以满足自身需求。

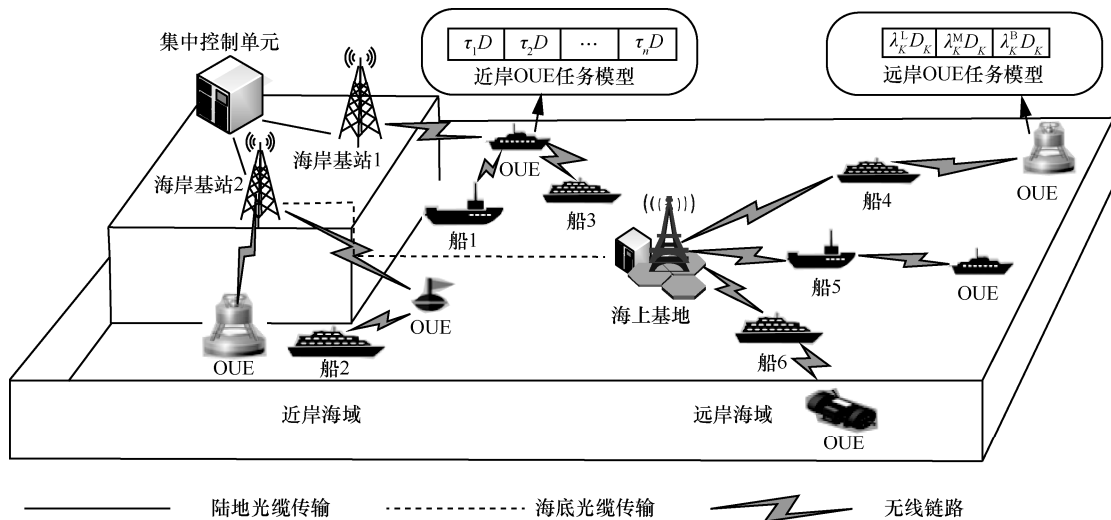


图 1 海洋网络系统模型

因此, 结合单用户模型特性, 近岸海域本文提出单用户多边缘节点模型。如图 1 所示, 该模型主要由集中控制单元、边缘计算节点 (包含海岸基站、大型船舶等) 以及 OUE (包含浮标、小型船只等) 组成。OUE 可以通过多个边缘计算节点协同处理计算任务, 调度近岸丰富的计算资源以满足自身需求。

多用户模型一般用于解决资源匮乏时如何实现资源的合理分配利用, 以满足多个用户的卸载需求。远岸海域计算资源匮乏, 在同一区域内多个用户同时卸载可用计算资源相对较少, 需要考虑有限资源的合理分配。此外, 远岸海域范围广, 节点离散化分布, OUE 可能远离海上基站 (OBS, offshore base station), 这使远离 OBS 的 OUE 需要多跳传输完成卸载。通常, 多于两跳的卸载由于时延与能耗开销巨大, 不符合低时延高可靠的海事应用需求。因此, 如图 1 所示, 本文面向远岸海域提出了双跳多用户模型, 主要由 OBS、中继节点 (大型船舶等组成)、 K 个 OUE 等组成。OUE 可将计算任务卸载至 OBS 和各自的中继节点, 从而在满足 OUE 卸载的同时, 实现资源的合理分配。

由于海洋网络节点间资源差异明显, OUE 计算卸载能耗与时延受到边缘节点资源影响。如 OUE 任务卸载至通信条件较好、计算能力较强的节点, 卸载时延较低。需要 OUE 选取合适的边缘计算节点使 OUE 在计算卸载时拥有良好的卸载条件十分必要。本文结合前期研究基础^[19]提出基于海洋网络连通概率的边缘计算节点选取方法, 该方案适用于近岸与远岸 2 种场景, 包含海洋网络连通概率预测和边缘计算节点确定 2 个步骤, 具体如下。

图 2 为相邻船舶通信示意。基于前期对海洋网络中节点间连通概率预测研究^[19], 船与船之间连通概率为

$$P_c = 1 - P(B=0, S(x, r)) = e^{-\rho_{\text{mob}} S(x, r)} \quad (1)$$

其中, ρ_{mob} 表示船舶密度 (艘/海里, 与船舶速度有关), $S(x, r)$ 表示船 A 与船 B 通信范围交集 (如图 2 中灰色区域所示), B 表示 $S(x, r)$ 内存在船舶的数量, $P(B=0, S(x, r))$ 为两船通信范围内没有船舶的概率, x 和 r 分别表示两船间距离和船舶通信半径, 且 x 、 r 与 $S(x, r)$ 存在如下关系

$$S(x, r) = 2r^2 \arccos \frac{x}{2r} - x \sqrt{r^2 - \frac{x^2}{4}}, \quad x \in [0, r] \quad (2)$$

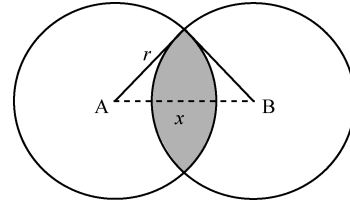


图 2 相邻船舶通信示意

图 3 为船舶与基站通信示意。船舶与基站间连通概率受两基站间距离 L 、基站覆盖半径 R 、船舶通信半径 r 及船舶到基站的距离 x 影响。结合图 3 给出了基站与船舶连通概率 P_s ^[19]。

- 1) 当 $0 < L \leq 2R$, $x \in [0, L]$ 时, $P_s = 1$ 。
- 2) 当 $2R < L \leq 2R + r$, $x \in (0, R) \cup (L - R, L)$ 时, $P_s = 1$ 。

- 3) 当 $2R < L \leq 2R + r$, $x \in (R, L - R)$ 时,

$$P_s = 1 - \frac{L - 2R}{L} \exp(-\rho_{\text{mob}}(2R + 2r - L)) \quad (3)$$

- 4) 当 $2R + r < L \leq 2R + 2r$, $x \in (0, R) \cup (L - R, L)$ 时, $P_s = 1$ 。

- 5) 当 $2R + r < L \leq 2R + 2r$, $x \in (R, L - R)$ 时,

$$P_s = 1 + \frac{2}{\rho_{\text{mob}} L} \left(e^{-\rho_{\text{mob}} r} - e^{-\rho_{\text{mob}}(2R + 2r - L)} \right) - \frac{2R + 2r - L}{L} e^{-\rho_{\text{mob}}(2R + 2r - L)} \quad (4)$$

- 6) 当 $L > 2R + 2r$ 时,

$$P_s = \frac{2R + 2r}{L} + \frac{2(e^{-\rho_{\text{mob}} r} - 1)}{\rho_{\text{mob}} L} \quad (5)$$

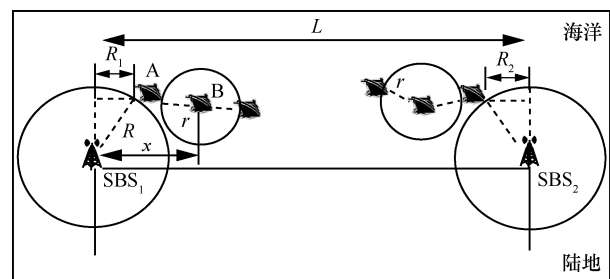


图 3 船舶与基站通信示意

根据以上船-船、船-基站之间连通概率, 可求得 OUE 与该区域内各节点之间的连通概率。将连通概率大于阈值 P_y 的节点作为连通节点, 可保障良好的计算卸载通信条件。

基于海洋网络连通概率预测模型, 选择网络中计算能力和通信资源较好的海洋连通节点作为 OUE 的边缘计算节点。可通过式(6)计算具体 OUE

与第 k 个连通节点之间的边缘系数 $S_k \in (0,1)$ ，根据每个连通节点的边缘系数与阈值 S_y 之间的关系确定边缘计算节点。

$$S_k = \mu \frac{E_k^r}{\sum_{k \in \text{oue}} E_k^r} + \eta \frac{\text{SNR}_{\text{oue}-k}}{\sum_{k \in \text{oue}} \text{SNR}_{\text{oue}-k}} + \gamma \frac{f_k}{\sum_{k \in \text{oue}} f_k} - \varphi \frac{d_{\text{oue}-k}}{\sum_{k \in \text{oue}} d_{\text{oue}-k}} \quad (6)$$

其中， $\mu + \eta + \gamma + \varphi = 1$ 。

当 $S_k \geq S_y$ 时，选取第 k 个连通节点作为 OUE 边缘计算节点。式(6)中第一部分表示第 k 个连通节点的剩余能量 E_k^r 占 OUE 所有连通节点能量和的比重，值越大表示该节点相对 OUE 的其他节点的能量优势越大；第二部分为 OUE 与第 k 个连通节点间的信噪比 $\text{SNR}_{\text{oue}-k}$ 在 OUE 所有连通节点中的占比，值越大表示该节点与 OUE 间的传输条件越好；第三部分表示第 k 个连通节点的计算能力 f_k 在 OUE 所有连通节点中的占比，值越大表示该节点相比 OUE 的其他节点计算能力越强；第四部分为 OUE 与第 k 个连通节点之间的距离 $d_{\text{oue}-k}$ 占 OUE 所有连通节点距离和的比重，值越小表示该节点与 OUE 间的数据传输能耗与时延越小。

4 基于多点协同的近岸计算卸载方法

4.1 近岸海域卸载流程描述

OUE 多应用并行执行（如自动定位系统、资源探测应用、娱乐应用等）会集中产生大量需要实时处理的数据。假设数据 D 可被分割成 N 个不同数据量的计算任务，表示为 $N = \{N_{\text{loc}}, N_{\text{off}}\} = \{\tau_1 D, \tau_2 D, \dots, \tau_n D\}$ ，其中 N_{loc} 和 N_{off} 分别表示在本地执行或计算卸载的任务数量； $\tau_n D$ 表示任务 n 需要处理的数据量， τ_n 表示任务 n 数据量分配比例且 $\sum_{n=1}^N \tau_n = 1$ 。

当 OUE 发送卸载请求至集中控制单元，控制单元通过获取 OUE 附近节点信息，制定最佳卸载方案。控制单元将卸载方案 $X_N = \{x_1, x_2, \dots, x_n\}$ （其中， x_n 表示任务 n 卸载至编号为 x_n 的边缘计算节点， $x_n = 0$ 表示本地执行）发送至 OUE。OUE 按照卸载策略将任务分别在本地处理或卸载至不同的边缘计算节点。各边缘计算节点将执行结果回传至 OUE。

当任务 n 在本地执行时，需要处理的数据量为 $\tau_n D$ bit。设本地计算能力为 f_{oue} （即每秒 CPU 周期数）， c_n 为计算 1bit 数据所需的 CPU 周期数，与任务复杂度有关。设 P_{loc} 为 OUE 的计算功率，因此任务 n 在本地处理所需时间和能耗可表示为

$$T_n^L = \frac{c_n \tau_n D}{f_{\text{oue}}} \quad (7)$$

$$E_n^L = P_{\text{loc}} T_n^L \quad (8)$$

当任务 n 卸载至边缘计算节点执行时，由于海洋观测监测传感网中各节点具有移动性且轨迹不同；OUE 在卸载时，首先需要考虑 OUE 与边缘计算节点间的有效通信时间，确保 OUE 在计算卸载时不会出现通信中断的问题。如图 4 所示，可设船 A 速度为 v_1 ，船 B 速度为 $v_2 = v_2 \cos \alpha + v_2 \sin \alpha$ （ α 为速度 v_2 与 i 轴的夹角），当船 B 进入船 A 有效通信范围时，直线 A—B 与 i 轴夹角为 β 。另设船 B 进入船 A 有效通信范围后沿 $\vec{P_1 P_2}$ 线路行驶，则船 A 与船 B 有效通信时间为

$$T_{A-B}^{\text{com}}(\alpha, \beta, v_1, v_2) = \frac{|2r \cos(\alpha + \beta)|}{|v_2 - v_1|} \quad (9)$$

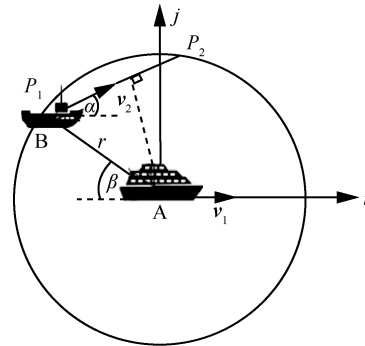


图 4 相邻两船通信示意

在保证卸载不会中断之后，任务 n 卸载时，设 $f_{\text{maec}}(x_n)$ 为边缘节点 x_n 的计算能力， P_{tran} 表示 OUE 传输功率，任务 n 卸载所需时间为

$$T_n^o = T_{\text{oue}-x_n} + T_{x_n}^c = \frac{\tau_n D}{r(x_n)} + \frac{c_n \tau_n D}{f_{\text{maec}}(x_n)} \quad (10)$$

其中， $T_{\text{oue}-x_n}$ 为任务 n 传输时间； $T_{x_n}^c$ 为任务 n 在边缘计算节点执行时间； $r(x_n)$ 为 OUE 与 x_n 间的传输速率， $r(x_n) = W \log(1 + (\text{SNR})_{\text{oue}-x_n})$ ， W 为 OUE 的传输带宽。根据任务 n 的传输时间 $T_{\text{oue}-x_n}$ ，任务 n 计算卸载所需能耗为

$$E_n^o = P_{\text{tran}} T_{\text{oue-x}_n} \quad (11)$$

近岸海域 OUE 的总执行时间 T_{oue} 受卸载决策 X_N 和任务分配系数 τ 影响。由于各任务之间可并行执行，卸载部分所用时间取决于边缘节点中执行时间的最大值。为了最小化 OUE 时延，近岸区域卸载时延优化目标如式(12)所示，其中第一部分表示本地计算部分的总执行时延，第二部分表示卸载部分的总执行时延。约束 a 表示任务卸载执行时延不大于任务 n 本地计算时延及 OUE 与边缘节点 x_n 的通信时间；约束 b 表示任务 n 卸载能耗不超过该任务本地计算能耗；约束 c 表示各任务的数据量之和应等于 OUE 需要处理的总数据量；约束 d 表示每个任务只能卸载至一个边缘节点。

$$\begin{aligned} \min T_{\text{oue}}(\tau, X_N) = & \\ \max \left\{ \sum_{n=1}^{N_{\text{loc}}} \frac{c_n \tau_n D}{f_{\text{oue}}}, \max \left(\frac{\tau_n D}{r(x_n)} + \frac{c_n \tau_n D}{f_{\text{maec}}(x_n)} \right) \right\} & \\ \text{s.t.} & \\ \text{a: } T_n^o \leq \min \{ T_n^L, T_{\text{oue-x}_n}^{\text{com}} \} & \\ \text{b: } E_n^o \leq E_n^L & \\ \text{c: } \sum_{n=1}^N \tau_n D = D, \tau_n \in (0, 1) & \\ \text{d: } \tau_n D \in x_n & \end{aligned} \quad (12)$$

4.2 近岸海域多节点协同卸载优化方法

为了解决近岸多节点协同机制下卸载优化问题，本文提出一种基于海洋多节点协同卸载的遗传算法 (MCO-GA, maritime multi-node cooperative offloading based genetic algorithm)。该算法通过对遗传算法 (GA, genetic algorithm) 中编码方式及适应度函数的优化与改进，使 OUE 在寻找到最优卸载策略的同时可有效降低时延。

4.2.1 遗传算法

GA 通过模拟自然进化过程来搜索最优解，染色体编码时每条染色体经过编码可以看作优化问题的一种解。染色体的编码影响后续的交叉、变异操作，很大程度上决定了 GA 的进化效率。适应度函数是根据目标函数确定的用于区分种群中个体好坏的标准。在染色体种群中，个体的适应度值越大，表明该个体越适应环境。

选择操作以一定概率从种群中选择若干个体。选择过程基于适应度优胜劣汰机制。通过计算，每个染色体被赋予一个适应度值，这些值与个体被选

择的概率相关。本文卸载优化策略采用轮盘赌选择方式^[20]，假设种群数量为 S ， A_1, A_2, \dots, A_i 表示当前种群中每个可能解，个体 i 的适应度为 $\text{fitness}(A_i)$ ，因此该个体被选中的概率为

$$P_i = \frac{\text{fitness}(A_i)}{\sum_i \text{fitness}(A_i)} \quad (13)$$

在进行交叉操作时，本文采用单点交叉的方式且交叉概率为 P_c ，具体操作是在个体编码串中随机设定一个交叉点，实行交叉时，该点前或后 2 个个体的部分结构互换，从而生成 2 个新个体。在变异操作中本文采用基本位变异方式，变异概率为 P_m ，随机指定一位或多位点做变异运算得到新个体，可防止算法过早收敛。

4.2.2 基于海洋多节点协同卸载的遗传算法

MCO-GA 通过对超出可行域的染色体增加惩罚值，使其适应度值减小，在选择操作时不易被选中；把近岸约束优化问题转化为无约束问题，最终实现在可行域内的最优计算卸载策略求解。

在对优化变量进行编码时，由于采用固定边缘节点与移动边缘节点协同卸载的策略；因此需要对 OUE 每个边缘节点进行编号后，将 OUE 各任务进行卸载的边缘计算节点编号及各边缘节点的任务分配比例采用二进制编码的方式表示在每条染色体上。其中，任务卸载的边缘计算节点编号及任务分配比例分别用卸载决策 X_N 和数据量分配比例 τ 表示。优化变量编码过程如图 5 所示， $x_1 - x_n$ ， $\tau_1 - \tau_n$ 分别表示任务卸载决策和任务分配比例。图 5 中列举了卸载决策变量 x_1 的二进制表示方法，其他变量同理。

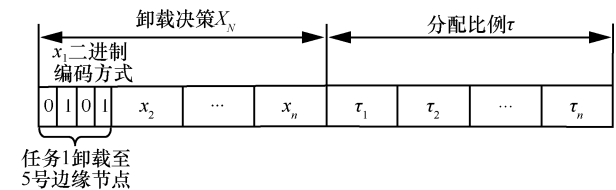


图 5 优化变量编码过程

由于 GA 面向非约束优化过程，而本文考虑了约束优化问题。因此，需要通过引入惩罚函数将约束优化问题转化为无约束优化问题进行分析，且适应度函数可表示为

$$\text{fitness}(A_i) = (T_{\text{oue}}(A_i) + \text{penalty}(A_i))^{-1} \quad (14)$$

其中， $\text{penalty}(A_i)$ 为惩罚函数（通过对非可行解的个体增加惩罚值使其被选择的概率降低），表示为

$$\text{penalty}(A_i) = l_1 \max\{0, T_n^o - \min\{T_n^L, T_{\text{oue}-x_n}^{\text{com}}\}\} + l_2 \max\{0, E_n^o - E_n^L\} \quad (15)$$

其中， l_1 和 l_2 为正惩罚因子。

结合海洋船舶运动轨迹的特殊性^[19]，需要考虑船舶间速度方向不同以及通信半径差异产生的通信时间变化问题。因此，惩罚函数考虑了有效通信时间（式(9)）等因素，从而避免了 OUE 计算卸载时通信中断问题。惩罚函数通过对卸载能耗进行约束，保证了 OUE 计算卸载时较低的能耗。

4.2.3 基于 MCO-GA 的优化策略

本文中每条染色体表示一种卸载策略 A_i ，其中每一种卸载策略 A_i 包括卸载决策 X_N 与数据量分配比例 τ 。此外，卸载策略 A_i 交叉变异操作前后的适应度可分别用 $\text{fitness}(A_i)$ 与 $\text{fitness}_n(A_i)$ 表示，适应度值包括该卸载策略下 OUE 对应的执行时延 $T_{\text{oue}}(A_i)$ 及相应的惩罚值 $\text{penalty}(A_i)$ 。图 6 描述了 MCO-GA 的流程，具体如下。

输入 边缘计算节点计算能力 $f_{\text{mac}}(x_n)$ 、OUE 计算能力 f_{ouc} 、任务数量 N 、OUE 本地计算功率 P_{loc} 、OUE 传输功率 P_{tran} 和任务 n 计算复杂度 c_n 。

输出 最优卸载决策 X_N 、最优数据量分配比例 τ 、最优卸载策略对应时延。

步骤 1 初始化。 确定迭代次数 itera_num 与染色体数量 S （每次迭代不同的卸载策略数目），同时设置卸载策略交叉操作与变异操作的概率分别为 P_c 和 P_m ，在可行域内随机产生 S 个卸载策略。

步骤 2 计算适应度。 根据式(14)的适应度函数计算每种卸载策略对应的适应度。

步骤 3 选择操作。 根据式(13)计算每种卸载策略 A_i 被选中的概率 P_i ，将本次迭代中所有可能的卸载策略按照其概率大小进行累加排序。同时均匀产生 S 个 $0 \sim 1$ 的随机数 P_r 。若 $P_i > P_r$ ，则选择该卸载策略进行后续操作，直至选出足够数量的卸载策略。

步骤 4 交叉操作。 每 2 种卸载策略一组，被赋予 $0 \sim 1$ 的随机数 P_{cr} （该值作为是否进行交叉操作的判断依据）。若 $P_{cr} < P_c$ ，则该组 2 种卸载策略进行交叉操作，产生 2 种新的卸载策略；否则 2 种策略保持不变。

步骤 5 变异操作。 每种卸载策略被赋予 $0 \sim 1$ 的随机数 P_{mr} （作为是否进行变异操作的判断依据）。若 $P_{mr} < P_m$ ，则随机选取该卸载策略上 3 个点位进行变异操作。

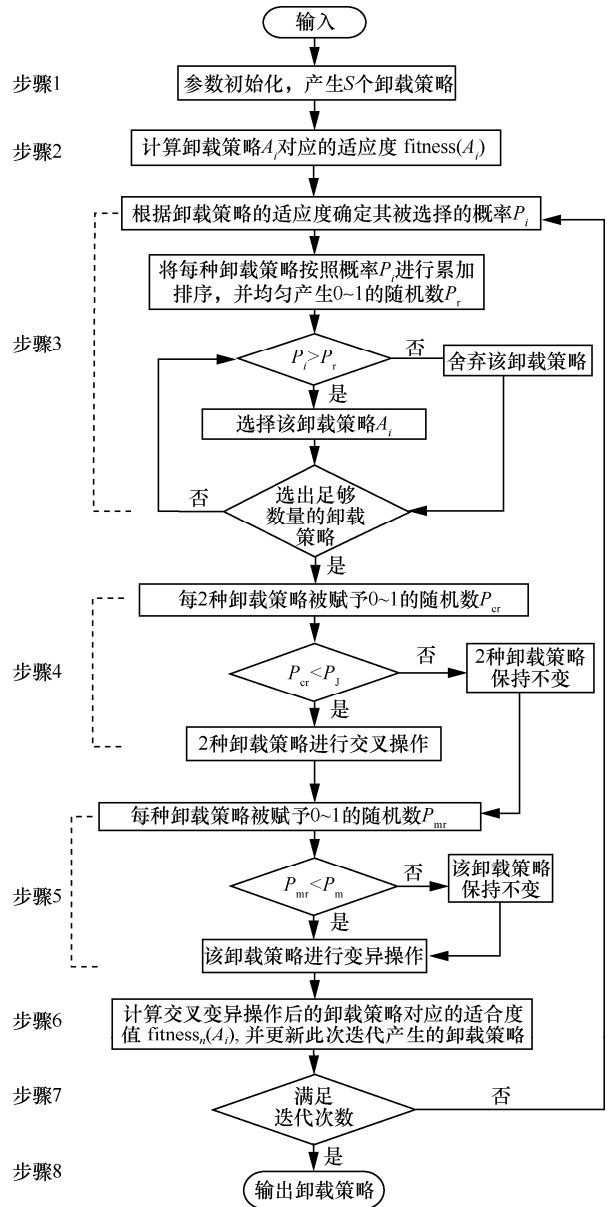


图 6 基于 MCO-GA 算法的优化策略流程

步骤 6 更新种群。 计算交叉变异操作后卸载策略 A_i 对应的适应度 $\text{fitness}_n(A_i)$ ，若 $\text{fitness}_n(A_i) < \text{fitness}(A_i)$ ，则替换交叉变异前的卸载策略 A_i 。

步骤 7 终止条件判断。 终止条件根据迭代次数 itera_num 判断。若满足迭代次数，则选择出适应度值最小的卸载策略并退出循环；否则继续执行步骤 3~步骤 6，直至满足条件。

步骤 8 输出最优卸载策略 A_i 及对应时延 $T_{\text{ouc}}(A_i)$ 。

5 基于容错重传机制的远岸计算卸载方法

5.1 远岸海域卸载流程描述

远岸海域计算资源相对匮乏且海域环境复杂, 节点间通信链路易受到恶劣环境因素的影响而导致卸载失败。提出容错重传机制计算卸载方法考虑节点间可能出现的通信故障十分必要, 可保证远岸海事应用的可靠性。

设远岸系统模型中继节点 $M = \{m_1, m_2, \dots, m_k\}$ (m_k 表示第 k 个 OUE 的中继节点), OBS 计算能力强于中继节点, 且 OBS 与每个中继节点均可作为边缘计算节点。另设 OUE 计算任务可分割, 每个 OUE 只能选择一个中继节点连接 OBS。一般情况下, K 个 OUE 可同时向控制单元发送卸载请求。控制单元收到请求后为 OUE 制定相应卸载策略。

$a_k = 0$ 表示第 k 个 OUE (OUE _{k}) 正常卸载, 此时, 卸载策略包含 OUE _{k} 最佳任务分配比例 $\lambda_k = \{\lambda_k^L, \lambda_k^M, \lambda_k^B\}$ (其中 λ_k^L 、 λ_k^M 、 λ_k^B 分别表示 OUE _{k} 在本地、 m_k 和 OBS 执行的任务比例) 和 OBS 的计算资源分配比例 θ_k 。 $a_k = 1$ 表示 m_k 与 OBS 之间传输失败, 需要将 $a_k = 0$ 时 OUE _{k} 分配给 OBS 的计算任务重新分配至 m_k 和本地进行处理。卸载策略包括重新规划的任务分配比例 $\lambda_k^{\text{re}} = \{\lambda_k^{\text{Lre}}, \lambda_k^{\text{Ore}}\}$, 其中, λ_k^{Lre} 、 λ_k^{Ore} 分别表示 OUE _{k} 在 OBS 卸载失败的任务在本地和 m_k 的任务分配比例。

举例说明, 当 $a_k = 0$ 时, OUE _{k} 可将计算任务 D_k 分别在本地、中继节点和 OBS 执行。本地执行需要处理的数据量为 $\lambda_k^L D_k$ 。设 OUE _{k} 本地计算能力为 f_k , P_k^L 为 OUE _{k} 计算功率, OUE _{k} 子任务在本地执行时延和能耗可表示为

$$T_k^L = \frac{c_k \lambda_k^L D_k}{f_k} \quad (16)$$

$$E_k^L = P_k^L T_k^L \quad (17)$$

OUE _{k} 全部任务在本地执行时延与能耗可表示

$$T_k^{\text{all}} = \frac{c_k D_k}{f_k} \quad (18)$$

$$E_k^{\text{all}} = P_k^L T_k^{\text{all}} \quad (19)$$

当 $a_k = 0$ 且 OUE _{k} 子任务在 m_k 执行时, 需要处理的数据量为 $\lambda_k^M D_k$ 。设中继节点 m_k 计算能力为

f_{m_k} , 子任务卸载至 m_k 执行时延可表示为

$$T_{m_k} = T_{m_k}^c + T_{k-m_k} = \frac{c_k \lambda_k^M D_k}{f_{m_k}} + \frac{(\lambda_k^M + \lambda_k^B) D_k}{r_{k-m_k}} \quad (20)$$

其中, $T_{m_k}^c$ 为任务在 m_k 的执行时延, T_{k-m_k} 为在 OBS 执行与在 m_k 执行的任务从 OUE _{k} 至 m_k 的传输时延, r_{k-m_k} 为 OUE _{k} 至 m_k 的传输速率。

当 $a_k = 0$ 且 OUE _{k} 子任务卸载至 OBS 执行时, 需要处理的数据量为 $\lambda_k^B D_k$ 。设 OBS 分配给 OUE _{k} 的计算资源为 $\theta_k f_{\text{OBS}}$ (f_{OBS} 表示 OBS 的计算能力), OUE _{k} 子任务卸载至 OBS 执行时延表示为

$$T_{B_k} = T_{B_k}^c + T_{k-m_k} + T_{m_k-B} = \frac{c_k \lambda_k^B D_k}{\theta_k f_{\text{OBS}}} + \frac{(\lambda_k^M + \lambda_k^B) D_k}{r_{k-m_k}} + \frac{\lambda_k^B D_k}{r_{m_k-B}} \quad (21)$$

其中, $T_{B_k}^c$ 为子任务在 OBS 执行时延, T_{k-m_k} 为在 OBS 执行与在 m_k 执行的子任务从 OUE _{k} 至 m_k 传输时延, T_{m_k-B} 为在 OBS 执行的子任务从 m_k 至 OBS 传输时延, r_{m_k-B} 为 m_k 与 OBS 间的传输速率。

因此, $a_k = 0$ 时, OUE _{k} 任务卸载部分 (包含 m_k 与 OBS 两部分) 的执行时延和能耗可表示为

$$T_k^O = \max\{T_{m_k}, T_{B_k}\} \quad (22)$$

$$E_k^O = P_k^{\text{tran}} T_{k-m_k} \quad (23)$$

其中, P_k^{tran} 为 OUE _{k} 的传输功率。

举例说明, 当 $a_k = 1$ 时, OUE _{k} 卸载至 OBS 失败, 此时需要将 $a_k = 0$ 时卸载至 OBS 的任务重新分割至本地和 m_k 处理。当 OUE _{k} 把未卸载成功的任务在本地执行时, 本地需要重新处理的数据量为 $\lambda_k^{\text{Lre}} \lambda_k^B D_k$ 。因此, 本地重新执行未卸载成功的部分任务所需时延和能耗可分别表示为

$$T_k^{\text{Lre}} = \frac{c_k \lambda_k^{\text{Lre}} \lambda_k^B D_k}{f_k} \quad (24)$$

$$E_k^{\text{Lre}} = P_k^L T_k^{\text{Lre}} \quad (25)$$

当 $a_k = 1$ 且未卸载成功的部分任务卸载至 m_k 时, 设 m_k 重新执行的数据量为 $\lambda_k^{\text{Ore}} \lambda_k^B D_k$, OUE _{k} 的传输速率不变, 重新卸载所需时延可表示为

$$T_k^{\text{Ore}} = T_{m_k}^{\text{rec}} + T_{k-m_k}^{\text{re}} = \frac{c_k \lambda_k^{\text{Ore}} \lambda_k^B D_k}{f_{m_k}} + \frac{\lambda_k^{\text{Ore}} \lambda_k^B D_k}{r_{k-m_k}} \quad (26)$$

其中, $T_{m_k}^{\text{rec}}$ 为任务在 m_k 重新处理时延, $T_{k-m_k}^{\text{re}}$ 为任务

重新传输至 m_k 的时延。OUE $_k$ 至 m_k 的任务重传所需能耗可表示为

$$E_k^{\text{Ore}} = P_k^{\text{tran}} T_{k-m_k}^{\text{re}} \quad (27)$$

综上所述, 容错机制下的计算卸载模型中, OUE $_k$ 总执行能耗与时延可分别表示为

$$E_k = E_k^{\text{L}} + E_k^{\text{O}} + a_k (E_k^{\text{Lre}} + E_k^{\text{Ore}}) \quad (28)$$

$$T_k = \max \left\{ T_k^{\text{L}} + a_k T_k^{\text{Lre}}, T_k^{\text{O}} + a_k (T_k^{\text{Ore}} + t_{\text{wait}}) \right\} \quad (29)$$

其中, t_{wait} 为 OUE $_k$ 的等待时延, 表示当 OUE $_k$ 在任务计划完成时间内未收到卸载任务的执行结果, 则启动容错机制, 先将原计划完成时间延长, 若还未收到 OBS 执行结果, 则认为任务卸载失败, 随后 $a_k = 0$ 时分配给 OBS 的任务做重传处理。由于不同 OUE 对于能耗和时延的敏感程度存在差异。在计算卸载时要综合考虑 OUE 对时延和能耗的要求, $I_{T,k}$ 和 $I_{E,k}$ 分别表示 OUE $_k$ 在综合优化时所要求的时延和能耗比重, 可根据 OUE 需求进行调整。OUE $_k$ 在执行任务时总开销 (时延和能耗) 为

$$C_k = I_{T,k} T_k + I_{E,k} E_k \quad (30)$$

其中, $I_{T,k} + I_{E,k} = 1$ 。

远岸计算资源匮乏区域可容错机制下, K 个 OUE 计算卸载受制于每个 OUE 正常执行与故障执行下的任务分配比, 以及正常执行情况下 OBS 的计算资源分配比。针对不同 OUE 的实际使用需求, 远岸容错机制下卸载开销优化问题可由式(31)表示, 其中 $\lambda = \{\lambda^{\text{L}}, \lambda^{\text{M}}, \lambda^{\text{B}}\}$ 且 $\lambda^{\text{L}}, \lambda^{\text{M}}, \lambda^{\text{B}}$ 分别为 K 个 OUE 在本地、中继节点、OBS 的任务分配比例集合。 $\lambda^{\text{re}} = \{\lambda^{\text{Lre}}, \lambda^{\text{Ore}}\}$, $\lambda^{\text{Lre}}, \lambda^{\text{Ore}}$ 分别表示重传情况下 K 个 OUE 在本地和中继节点的任务分配比例。约束 a 表示 OUE $_k$ 任务执行总时延不超过节点间有效通信时间及任务全部本地执行的最小值; 约束 b 表示 OUE $_k$ 任务执行能耗不大于任务全部本地执行的能耗; 约束 c 表示正常执行和重传执行的任务分配比例取值范围; 约束 d 表示正常执行情况下, 任务分配比例的和约束; 约束 e 表示重传情况下, 任务分配比例的和约束; 约束 f 表示 OBS 分配给 K 个 OUE 计算资源不大于 OBS 计算资源。

$$\min C(\lambda, \theta, \lambda^{\text{re}}) = \min \sum_{k=1}^K (I_{T,k} T_k + I_{E,k} E_k)$$

$$\text{a: } T_k \leq \min \{ T_{k-m_k}^{\text{com}}, T_{m_k-B}^{\text{com}}, T_k^{\text{all}} \}$$

$$\text{b: } E_k \leq E_k^{\text{all}}$$

$$\text{c: } \lambda, \lambda^{\text{re}} \in [0, 1]$$

$$\text{d: } \lambda_k^{\text{L}} + \lambda_k^{\text{M}} + \lambda_k^{\text{B}} = 1$$

$$\text{e: } \lambda_k^{\text{Lre}} + \lambda_k^{\text{Ore}} = 1$$

$$\text{f: } \sum_{k=1}^K \theta_k f_{\text{OBS}} \leq f_{\text{OBS}} \quad (31)$$

5.2 远岸海域可容错卸载优化方法

为解决远岸计算卸载优化问题, 本文提出基于分组交叉学习的粒子群优化 (GCL-PSO, group cross learning based particle swarm optimization) 算法。由于此时优化变量和问题约束条件增多, 传统群智能算法在求解问题时容易陷入局部最优。GCL-PSO 算法在传统 PSO 算法基础上引入分组竞争学习, 通过每个粒子分组交叉竞争学习, 使其能够全面地在可行域内搜索最优解。GCL-PSO 算法较一般算法收敛速度快, 处理复杂问题时能够实现快速优化。

5.2.1 传统 PSO 算法

PSO 算法设置粒子总数为 `particle_num`, 包含位置与速度因素^[21]。第 i 个粒子的位置可看作优化问题的一种潜在解 Q_i , 第 i 个粒子的速度 h_i 表示粒子 i 位置更新的方向与距离。寻优过程中粒子 i 搜索到自身最优的位置表示为 `pbest $_i$` , 粒子群中搜索到群体最优的位置表示为 `gbest`。粒子 i 位置与速度更新计算过程表示为

$$h_i(j+1) = \omega h_i(j) + c_1 r_1 (\text{pbest}_i(j) - Q_i(j)) + c_2 r_2 (\text{gbest}(j) - Q_i(j)) \quad (32)$$

$$Q_i(j+1) = Q_i(j) + h_i(j+1) \quad (33)$$

其中, j 表示迭代次数, ω 为惯性因子 (值较大时全局寻优较强, 值较小时局部寻优能力较强); c_1, c_2 为学习因子; r_1 和 r_2 为 0~1 的随机数。

5.2.2 基于分组交叉学习的粒子群优化算法

本文为远岸卸载优化问题提出 GCL-PSO 算法。结合传统 PSO 算法中调节参数少、收敛速度快等优点。所提算法先对每个粒子进行分组, 使其在小组内先进行局部学习, 再通过粒子更新式进行全局学习。下面将介绍 GCL-PSO 算法的具体实现过程。

GCL-PSO 算法初始化粒子参数后, 将每个粒子进行随机分组。分组后根据适应度大小在组内排序。组内每个粒子依据排序顺序在组内相互学习, 学习方式采用 GA 的交叉操作。组内适应度较差的

粒子通过向适应度较好的粒子学习获取新粒子，并将新粒子的适应度与交叉前的粒子适应度比较。若适应度优于交叉前的粒子，则将交叉前的粒子替换为交叉后的粒子，再将新粒子与当前个体极值与全局极值比较。若适应度优于当前个体极值或全局极值则将其替换，否则舍弃新粒子。

分组学习后将每个粒子从分组中拆分出来（下次迭代重新分组），根据分组学习后输出的 $pbest_i$ 与 $gbest$ ，更新式(32)和式(33)进行全局学习，通过不断调整每个粒子位置与速度最终找到优化问题的最优解。

5.2.3 基于 GCL-PSO 算法的优化策略

GCL-PSO 算法相关参数与远岸卸载优化问题相关参数间的映射关系如表 1 所示。

表 1 GCL-PSO 算法参数与卸载策略参数映射

GCL-PSO 算法	远海可容错优化问题
粒子位置	卸载策略
适应度值	系统总成本
种群	不同卸载策略集合

表 1 中粒子位置结合卸载问题可表示为 $Q_i(j) = \{Q_i^N(j), Q_i^F(j)\}$ ，其中， $Q_i^N(j)$ 表示正常情况下第 j 次迭代时的卸载策略， $Q_i^F(j)$ 表示故障情况下第 j 次迭代时的卸载策略。此外，粒子 i 第 j 次迭代时分组学习前后系统总成本可分别表示为 $C_i(j)$ 和 $C_i^c(j)$ 。图 7 给出了基于 GCL-PSO 算法的优化策略流程，具体如下。

输入 OUE 数量 K 、OUE 计算能力 f_k 、中继节点计算能力 f_{m_k} 、基站计算能力 f_{obs} 、OUE 本地计算功率 P_k^L 、OUE 传输功率 P_k^{tran} 、OUE 传输带宽 W_k 、中继节点传输带宽 W_{m_k} 和 OUE 任务复杂度系数 c_k 。

输出 最优卸载策略和最优策略对应系统成本。

步骤 1 初始化。确定迭代次数 GCL_itera 与粒子数量 $particle_num$ （每次迭代产生的卸载策略数量）在可行域内设置每个粒子初始位置 $Q_i^N(j)$ 与速度 $h_i(j)$ 。

步骤 2 系统成本计算。根据式(31)所示的优化问题计算正常情况下每种卸载策略 $Q_i^N(j)$ 所对应的系统总成本 $C_i(j)$ 。将前 j 次迭代中粒子 i 系统成本最小时所对应的卸载策略设置为粒子 i 当前的个体极值 $pbest_i(j)$ ，同时设置当前粒子群中系统成本最

小的粒子所对应的卸载策略为全局最优卸载策略 $gbest(j)$ 。

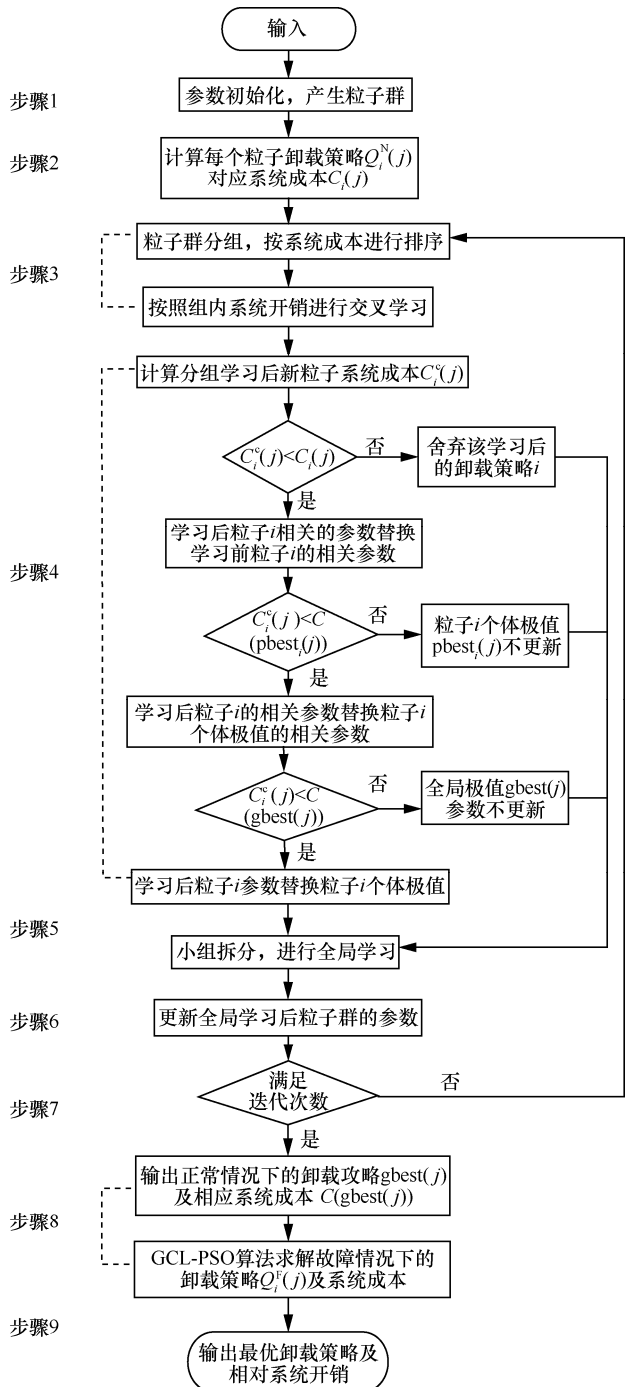


图 7 GCL-PSO 优化方法流程

步骤 3 卸载策略分组竞争学习。每 3 种卸载策略为一组进行随机分组。在组内按照每种策略所求得系统成本 $C_i(j)$ 对卸载策略进行排序，排序后系统成本值最大的策略采用交叉方式向组内次优策略学习，次优策略向成本值最小的

策略学习。

步骤 4 更新分组学习后参数。根据交叉学习后新粒子的卸载策略计算该策略下的系统总成本 $C_i^c(j)$ 。若 $C_i^c(j) < C_i(j)$, 则替换交叉学习前粒子对应的卸载策略; 若 $C_i^c(j) < C(\text{pbest}_i(j))$, 则更新该粒子个体极值 $\text{pbest}_i(j)$ 及对应的系统成本; 若 $C_i^c(j) < C(\text{gbest}(j))$, 则更新全局极值 $\text{gbest}(j)$ 及对应的系统成本。

步骤 5 粒子全局学习。根据式(32)和式(33)更新粒子 i 当前速度 $h_i(j)$ 与卸载策略 $Q_i^N(j)$ 。每个粒子根据当前个体极值 $\text{pbest}_i(j)$ 与全局极值 $\text{gbest}(j)$ 不断调整自身卸载策略。

步骤 6 更新全局学习后参数。求解全局学习后新粒子对应的系统总成本, 根据系统成本更新每个粒子的个体极值 $\text{pbest}_i(j)$ 和全局极值 $\text{gbest}(j)$ 。

步骤 7 终止条件判断。终止条件根据迭代次数进行判断。若满足迭代次数则退出循环, 输出正常执行时的最优的卸载策略 $\text{gbest}(j)$ 以及对应系统成本 $C(\text{gbest}(j))$ 。若不满足则继续执行步骤 3~步骤 6, 直至满足终止条件。

步骤 8 故障情况下最优策略求解。根据步骤 7 输出正常情况下最优卸载策略 $\text{gbest}(j)$, 运用 GCL-PSO 算法求解故障情况下的卸载策略 $Q_i^F(j)$ 。

步骤 9 输出最优卸载策略及对应的系统成本。输出每个 OUE 的卸载策略 (包含正常执行与故障执行 2 种情况下的策略) 以及对应的系统总成本。

6 实验仿真结果与分析讨论

6.1 近岸仿真结果分析

模拟 OUE 在近岸场景下同时将多个任务卸载至固定边缘计算节点 (基站) 和移动边缘计算节点 (船舶) 的网络场景。实验仿真参数设置参考文献[4,17,19], 如表 2 所示。为保证数值选取的均匀随机性, 通信半径的设置基于文献[19]并进行了相应的调整; 船舶速度的设置也主要参考文献[19]中的研究结果 (该研究结论说明海洋网络中船舶速度与单位区域内船舶密度呈负相关, 且近岸区域网络节点密度一般高于远岸区域)。因此, 在表 2 所示仿真实验场景下, 设置近岸船舶速度大于远岸船舶速度。此外, 通过多次实验, MCO-GA 在迭代 100 次以内均可实现稳定收敛, 因此设迭代次数

$\text{itera_num} = 100$, MCO-GA 中种群数量 $S = 70$, 交叉概率为 $P_c = 0.6$, 变异概率为 $P_m = 0.07$, 惩罚函数中惩罚因子 $l_1 = 2$, $l_2 = 1.5$, 变异点数目为 3。

表 2 近岸实验仿真参数

参数描述	参数值
任务数量 N	8
传输带宽 W /MHz	1
OUE 计算能力 f_{ouc} /GHz	1
移动边缘节点计算能力/GHz	[1, 3]
固定边缘节点计算能力/GHz	[6, 7]
OUE 计算功率 P_{loc} /W	1
OUE 传输功率 P_{tran} /W	0.4
任务复杂度 c_n /(cycle-bit ⁻¹)	800
信噪比 SNR /dB	[30, 50]
船舶通信半径 r /m	[100, 300]
基站通信半径 R /m	1 000
船舶速度 v /Knot	[15, 20]

为验证 MCO-GA 在卸载方面的有效性, 分析 MCO-GA 在不同数据量下的收敛效果以及在降低 OUE 时延方面的性能, 并将 MCO-GA 与以下卸载策略进行比较^[4,22]。1) 基站卸载策略 (简称为 BSprocess 策略), 总任务平均后卸载至多个基站处理, 而不卸载至移动边缘节点; 2) 随机卸载 (简称为 RANDOM 策略), 各任务数据量与边缘计算节点随机分配; 3) 本地执行 (简称为 LOCAL 策略), 任务全部在本地执行, 不进行卸载; 4) 将文献[22]中所提的蚁群优化 (ACO, ant colony optimization) 应用至近岸多节点协同的卸载策略, 简称为 ACO 策略。

为了验证 MCO-GA 能够在有限的迭代次数内实现稳定收敛, 本文对 MCO-GA 在不同数据量下的迭代收敛效果进行了验证。如图 8 所示, 随着迭代次数的逐渐增大, MCO-GA 在不同数据量下得到的时延解逐渐减小, 在迭代约 30 次时, 可寻找到其对应情况下的最优解实现稳定收敛 (部分情况在迭代 100 次以内实现收敛, 如数据量为 1 000 kB 时, 迭代约 60 次实现收敛)。通过分析, 可以看出 MCO-GA 在不同数据量下均可实现有限迭代次数内的稳定收敛, 且收敛速度较快, 使算法的运算量相对较低。此外, MCO-GA 时间复杂度与其他群智能算法相比较低^[11,17], 运行时间较短, 可在卸载精度要求不严苛的环境下快速为 OUE 找到一个可行解, 具有一定的实际应用性。

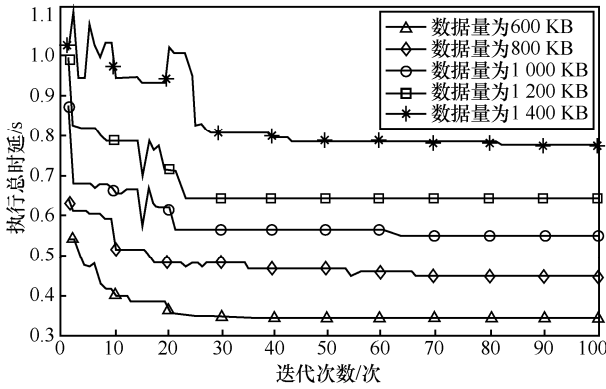


图 8 MCO-GA 迭代收敛

图 9 说明 5 种策略执行时延总体变化趋势均随着 OUE 数据量的增大而增加，但 MCO-GA 策略时延增加最少。这是由于 MCO-GA 结合近岸海洋特色对传统 GA 进行改进，使其更加适应近岸多点协同卸载问题的寻优。而基于 ACO 算法的策略由于寻优时容易陷入局部最优，其执行时延在不同数据量下均大于 MCO-GA，同时证明了 MCO-GA 较好的寻优效果。由于仿真设置了相同的基站与船舶数目使 BSprocess 策略执行时延相对较小，若减少基站数目，BSprocess 策略执行时延会随着数据量的增加而大幅增加。由于 RANDOM 策略的随机性，在不同数据量下的执行时延波动较大，数据量为 1 200 KB 的执行时延明显高于 1 400 KB 的情况。受 OUE 本地计算能力的限制，LOCAL 策略在不同数据量下执行时延最高。通过对比可以看出，MCO-GA 在降低 OUE 执行时延方面具有最好性能，在不同情况下均具有稳定收敛效果。

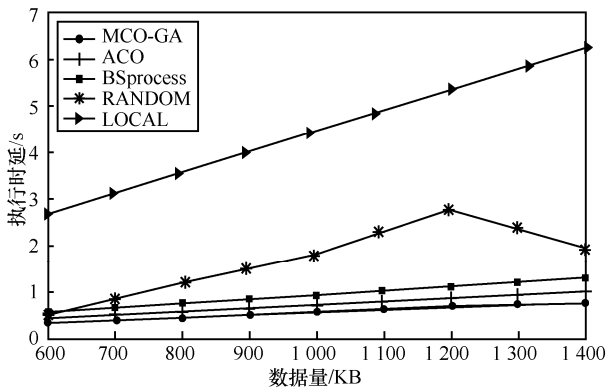


图 9 不同数据量下各策略时延对比

当移动边缘节点计算能力保持不变时，随着固定边缘节点计算能力的增加，不同边缘节点计算能力所对应时延仿真结果如图 10 所示（以固定数据量为 800 KB 为例）。可以看出，随着固定边缘节点

计算能力的提升，4 种策略任务执行时延总体均有不同程度的降低，提高了 OUE 任务执行效率。MCO-GA 在不同的计算能力下均具有最小时延。再次表明所提算法随着固定边缘节点计算能力的变化可以灵活地调整卸载策略以确保系统的实时性，并且具有更好的动态适应性。

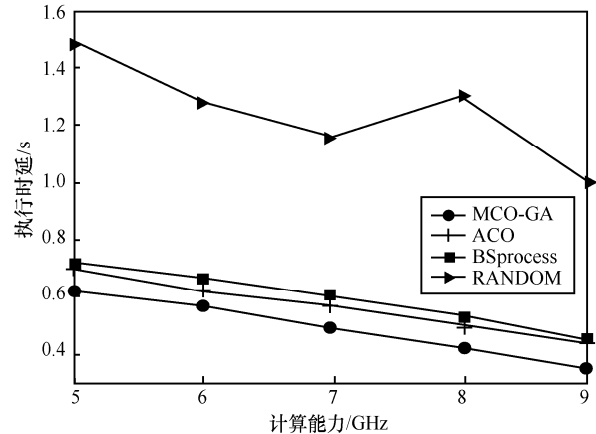


图 10 不同计算能力下各策略时延对比（固定数据量为 800 KB）

6.2 远岸仿真结果分析

远岸场景下模拟 K 个 OUE 通过各自中继节点与海上基站进行卸载的网络场景，仿真参数如表 3 所示^[4,19,23]。设置粒子数量 `particle_num = 60`，迭代次数 `GCL_itera = 100`，粒子更新速度 h_t 取值范围为 $[-0.25, 0.25]$ ，惯性因子 $\omega = 0.4$ ，随机数 $r_1 = 0.4$ ， $r_2 = 0.6$ 。为验证基于 GCL-PSO 算法的卸载策略在远岸容错机制下卸载有效性，将 GCL-PSO 与以下策略进行比较。1) PSO 策略，将传统的 PSO 算法应用于远岸的卸载策略；2) 平均卸载策略（简称为 AVERAGE 策略），将每个 OUE 的计算任务平均分配至本地，中继节点以及 OBS，OBS 计算资源平均

表 3 远岸实验仿真参数

参数描述	参数值
OUE 计算能力 f_k /GHz	[1, 1.5]
中继节点计算能力 f_m /GHz	[2, 3.5]
海上基站计算能力 f_{OBS} /GHz	12
OUE 计算功率 P_k^l /W	[0.8, 1]
OUE 传输功率 P_k^{tran} /W	[0.3, 0.5]
OUE 任务复杂度 c_k /(cycle-bit ⁻¹)	[800, 1 000]
船舶通信半径 r /m	[100, 300]
基站通信半径 R /m	1 200
船舶速度 v /Knot	[20, 30]

分配给 K 个 OUE; 3) 改进 PSO 策略, 将文献[21]中的改进 PSO 算法应用于远岸卸载策略; 4) 将文献[24]所提 AFSA 人工鱼群算法 (AFSA, artificial fish swarms algorithm) 应用于远岸的卸载策略, 简称为 AFSA 策略。

为了验证基于 GCL-PSO 算法的策略在解决远岸卸载问题时的寻优和收敛效果, 本文将 GCL-PSO 策略与其他策略进行对比。如图 11 所示, 当 OUE 数量 $K=6$ 且计算任务量为 500~900 KB 时, 基于 4 种策略求得的卸载总成本随着迭代次数的增加趋于稳定, 且均可在较少的迭代次数 (一般为 10~20 次) 内实现快速收敛, 运算量小。但通过对比 4 种策略的寻优效果可知, GCL-PSO 策略寻优得到的系统总成本远小于其他 3 种策略, 这展示了该策略良好的寻优效果。综上所述, 本文所提基于 GCL-PSO 算法的策略在解决远岸卸载问题时可在较少的迭代次数内实现稳定收敛且寻优效果远好于其他策略。同时, GCL-PSO 策略时间复杂度为 $O(mn)$ (m 和 n 分别代表迭代次数与粒子数目), 迭代次数少, 算法运算量小, 说明所需运行时间较短, 可满足实际应用需求。

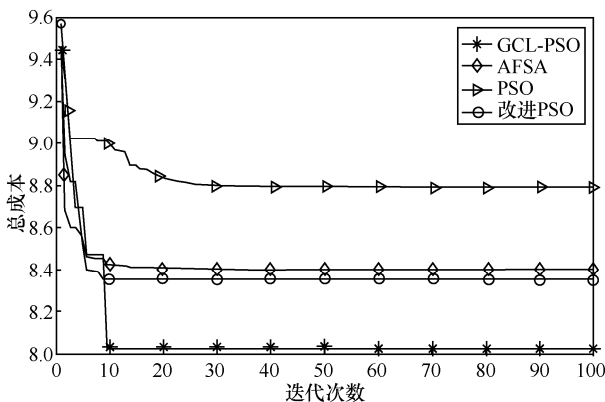


图 11 算法寻优和收敛效果

本文同时对比了系统中 OUE 均可正常卸载时各种算法的总成本, 如图 12 所示。可以发现随着系统中 OUE 数量的增加, 5 种策略的计算卸载系统总成本均随之增加, 但 GCL-PSO 的总成本较其他 4 种策略增幅较小。这是由于 GCL-PSO 引入了分组学习使在远岸多优化变量的复杂情况下仍能制定最优的计算卸载策略, 从而降低成本。由于寻优时容易陷入局部最优, 改进 PSO、AFSA、PSO 这 3 种策略较 GCL-PSO 策略相比卸载成本依次增加。随着 OUE 数量的增加 AVERAGE 策略系统总

成本相对增加, 虽然波动较小, 但未实现资源的合理调配, 使系统总成本较前 4 种策略更大。综上所述, 可以发现在正常卸载时, GCL-PSO 能够实现计算卸载最佳决策, 并可高效地降低系统时延与能耗。

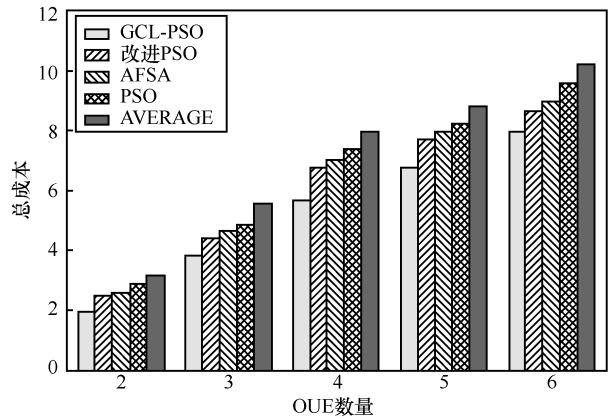


图 12 正常情况下不同 OUE 数量成本对比

设置系统中故障 OUE 数量为 1, 本文对比了故障情况下不同 OUE 数量的计算卸载总成本, 如图 13 所示。从图 13 可以看出, 随着 OUE 数量的增加, 5 种方案整体成本逐渐增加, 但本文所提方案卸载总成本相对最低。同时, 可以发现, 当 OUE 数量大于 4 时, 所提方案卸载成本远小于其他方案卸载成本, 表明故障情况下所提方案在 OUE 数量较多的系统中能有效地降低系统时延和能耗, 可以很好地适应多 OUE 系统中故障情况的计算卸载。综上所述, 随着 OUE 数量的变化, 所提 GCL-PSO 策略在正常与故障情况下均可实现最优卸载决策, 可以很好地处理故障情况下多 OUE 的计算卸载问题。

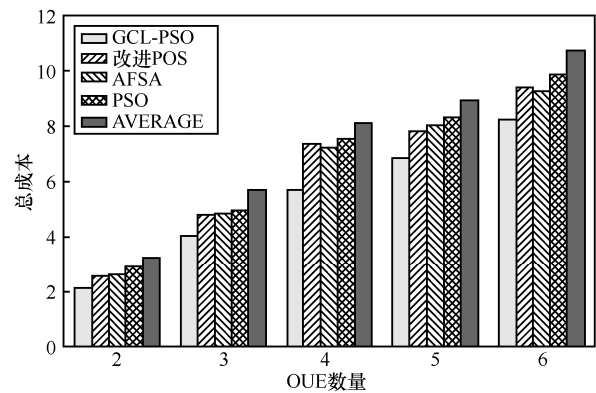


图 13 故障情况下不同 OUE 数量成本对比

图 14 对正常情况下 5 种方法在不同数据量下的卸载成本进行了分析, 实验设置 OUE 数量为 5。

从图 14 可以看出, 数据量的增加使 5 种策略的卸载成本均有不同程度的上升, 但基于 GCL-PSO 策略得到的卸载成本最小, 表明 GCL-PSO 策略能够很好地适应不同数据量下的卸载策略寻优。此外, 当数据量为 500~600 KB 时, GCL-PSO 策略较改进 PSO 策略相比成本降低了 20%; 当数据量为 600~700 KB 时, GCL-PSO 策略比改进 PSO 策略降低了 24% 的卸载成本。上述结果表明, 随着数据量的增加, GCL-PSO 策略在降低卸载成本方面的优势较更加明显。

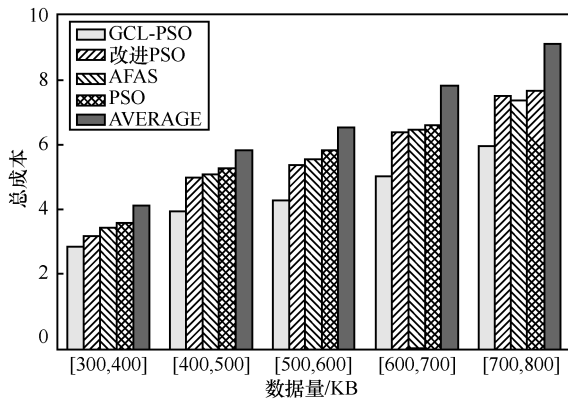


图 14 正常情况下不同数据量成本对比

图 15 对比了故障情况下 5 种策略的总成本。实验假设 OUE 数量为 5, 且其中一个 OUE 在卸载时其对应中继和 OBS 传输出现故障, 需要重传卸载。对比 5 种策略在不同数据范围下卸载总成本, 图 15 说明随着 OUE 任务数据量增加, 5 种策略系统总成本均增加, 但 GCL-PSO 能在故障情况下保持较低系统成本, 且系统成本增幅较小。这是由于该策略在制定正常执行时最佳卸载策略的基础上, 分析并提前制定了故障情况下任务重传的卸载策略。而其他 4 种策略由于在正常执行时寻优能力不够, 使在此基础上的故障重传分析寻优结果不准确, 成本较高。

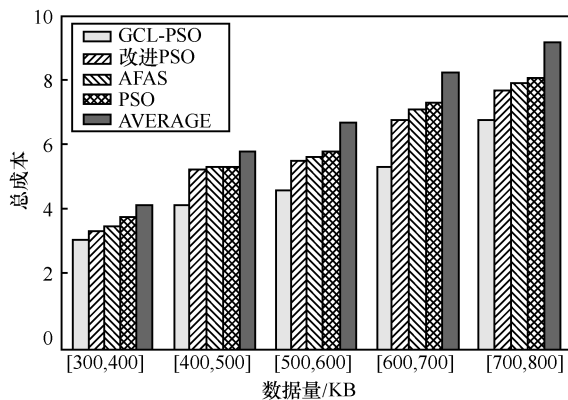


图 15 故障情况下不同数量成本对比

7 结束语

本文建立了近岸与远岸 2 种符合海洋特色的卸载模型。近岸场景下提出了多节点协同的卸载策略, 并针对该策略提出了 MCO-GA 以进一步降低 OUE 卸载时延。远岸场景下提出了 GCL-PSO 算法, 使远岸容错卸载策略能够找到最优的卸载策略。仿真结果表明, 在近岸场景下, 所提算法在不同数据量下均能实现稳定收敛, 且较传统方案能够极大地降低 OUE 时延。在远岸场景下, GCL-PSO 算法能够实现快速收敛, 与其他算法相比更加符合远岸容错机制下的卸载策略。

针对 PSO 算法中局部寻优能力差和寻优片面的缺陷, GCL-PSO 算法通过引入分组竞争学习思想对 PSO 算法进行了改进。仿真结果表明, GCL-PSO 算法较传统 PSO 算法寻优能力有较大提升。未来, 将针对传统粒子群收敛精度低等缺陷对 PSO 算法中的惯性因子 ω 等参数进行调整, 可进一步提高传统 PSO 算法的收敛精度。此外, 将进一步分析海洋环境因素对卸载的影响, 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监测能力奠定坚实的基础。

参考文献:

- [1] ZHOU Y Q, LIU L, WANG L, et al. Service-aware 6G: an intelligent and open network based on the convergence of communication, computing and caching[J]. Digital Communications and Networks, 2020, 6(3): 253-260.
- [2] ZHOU Y Q, TIAN L, LIU L, et al. Fog computing enabled future mobile communication networks: a convergence of communication and computing[J]. IEEE Communications Magazine, 2019, 57(5): 20-27.
- [3] MACH P, BECVAR Z. Mobile edge computing: a survey on architecture and computation offloading[J]. IEEE Communications Surveys & Tutorials, 2017, 19(3): 1628-1656.
- [4] XIAO Y, REN Z Y, ZHANG H L. A novel task allocation for maximizing reliability considering fault-tolerant in VANET real time systems[C]//2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications. Piscataway: IEEE Press, 2017: 1-7.
- [5] SAFAVAT S, SAPAVATH N N, RAWAT D B. Recent advances in mobile edge computing and content caching[J]. Digital Communications and Networks, 2020, 6(2): 189-194.
- [6] 谢人超, 廉晓飞, 贾庆民, 等. 移动边缘计算卸载技术综述[J]. 通信学报, 2018, 39(11): 138-155.
XIE R C, LIAN X F, JIA Q M, et al. Survey on computation offloading in mobile edge computing[J]. Journal on Communications, 2018, 39(11): 138-155.

- [7] QI Y L, TIAN L, ZHOU Y Q, et al. Mobile edge computing-assisted admission control in vehicular networks: the convergence of communication and computation[J]. IEEE Vehicular Technology Magazine, 2019, 14(1): 37-44.
- [8] ZHENG K, ZHENG Q, CHATZIMISIOS P, et al. Heterogeneous vehicular networking: a survey on architecture, challenges, and solutions[J]. IEEE Communications Surveys & Tutorials, 2015, 17(4): 2377-2396.
- [9] LIU J, MAO Y Y, ZHANG J, et al. Delay-optimal computation task scheduling for mobile-edge computing systems[C]//2016 IEEE International Symposium on Information Theory. Piscataway: IEEE Press, 2016: 1451-1455.
- [10] KAMOUN M, LABIDI W, SARKISS M. Joint resource allocation and offloading strategies in cloud enabled cellular networks[C]//2015 IEEE International Conference on Communications. Piscataway: IEEE Press, 2015: 5529-5534.
- [11] DENG M F, TIAN H, FAN B. Fine-granularity based application offloading policy in cloud-enhanced small cell networks[C]//2016 IEEE International Conference on Communications Workshops. Piscataway: IEEE Press, 2016: 638-643.
- [12] WANG H S, LI X, JI H, et al. Dynamic offloading scheduling scheme for MEC-enabled vehicular networks[C]//2018 IEEE/CIC International Conference on Communications in China. Piscataway: IEEE Press, 2018: 206-210.
- [13] SHA K W, YANG T A, WEI W, et al. A survey of edge computing-based designs for IoT security[J]. Digital Communications and Networks, 2020, 6(2): 195-202.
- [14] WEI F, CHEN S X, ZOU W X. A greedy algorithm for task offloading in mobile edge computing system[J]. China Communications, 2018, 15(11): 149-157.
- [15] 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.
- [16] GUO F X, ZHANG H L, JI H, et al. An efficient computation offloading management scheme in the densely deployed small cell networks with mobile edge computing[J]. IEEE/ACM Transactions on Networking, 2018, 26(6): 2651-2664.
- [17] 刘锐. 基于移动边缘计算的任务卸载策略研究[D]. 上海: 东华大学, 2019.
LIU R. Research on the task offloading strategy based on mobile edge computing[D]. Shanghai: Donghua University, 2019.
- [18] ZHAO H Y, WANG Y, SUN R J. Task proactive caching based computation offloading and resource allocation in mobile-edge computing systems[C]//2018 14th International Wireless Communications & Mobile Computing Conference. Piscataway: IEEE Press, 2018: 232-237.
- [19] SU X, MENG L L, HUANG J. Intelligent maritime networking with edge services and computing capability[J]. IEEE Transactions on Vehicular Technology, 2020, 69(11): 13606-13620.
- [20] 陈卓, 冯钢, 刘怡静, 等. MEC 中基于改进遗传模拟退火算法的虚拟网络功能部署策略[J]. 通信学报, 2020, 41(4): 70-80.
CHEN Z, FENG G, LIU Y J, et al. Virtual network function deployment strategy based on improved genetic simulated annealing algorithm in MEC[J]. Journal on Communications, 2020, 41(4): 70-80.
- [21] 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 computing offloading model based on reputation value in mobile edge computing[J]. Journal on Communications, 2020, 41(7): 141-151.
- [22] WANG M Y, JIN H, ZHAO C L, et al. Delay optimization of computation offloading in multi-hop ad hoc networks[C]//2017 IEEE International Conference on Communications Workshops. Piscataway: IEEE Press, 2017: 314-319.
- [23] 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 and Networks, 2019, 5(1): 10-17.
- [24] ZHANG H L, GUO J, YANG L C, et al. Computation offloading considering fronthaul and backhaul in small-cell networks integrated with MEC[C]//2017 IEEE Conference on Computer Communications Workshops. Piscataway: IEEE Press, 2017: 115-120.

[作者简介]



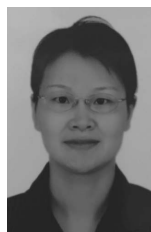
苏新(1986-), 男, 河北霸州人, 博士, 河海大学教授、硕士生导师, 主要研究方向为通信与计算融合、移动通信技术、边缘/雾计算、智慧海洋等。



薛淦阳(1996-), 男, 山西临汾人, 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。



周一青(1975-), 女, 浙江杭州人, 中国科学院计算技术研究所研究员、博士生导师, 主要研究方向为通信与计算融合、移动边缘计算、存储通信、干扰管控等。



朱金秀(1972-), 女, 江苏常州人, 博士, 河海大学副教授、硕士生导师, 主要研究方向为神经网络、边缘/雾计算、数字通信等。