



## 基于多群体混合智能优化算法的卸载决策寻优方法

方浩添, 田乐, 郭茂祖

引用本文:

方浩添, 田乐, 郭茂祖. 基于多群体混合智能优化算法的卸载决策寻优方法[J]. *智能系统学报*, 2024, 19(6): 1573–1583.

FANG Haotian, TIAN Le, GUO Maozu. Unloading decision optimization method based on multi-population hybrid intelligent optimization algorithm[J]. *CAAI Transactions on Intelligent Systems*, 2024, 19(6): 1573–1583.

在线阅读 View online: <https://dx.doi.org/10.11992/tis.202312042>

## 您可能感兴趣的其他文章

### 基于变步长蚁群算法的移动机器人路径规划

Mobile robot path planning based on variable-step ant colony algorithm

智能系统学报. 2021, 16(2): 330–337 <https://dx.doi.org/10.11992/tis.202004011>

### 非光滑凸情形Adam型算法的最优个体收敛速率

Optimal individual convergence rate of Adam-type algorithms in nonsmooth convex optimization

智能系统学报. 2020, 15(6): 1140–1146 <https://dx.doi.org/10.11992/tis.202006046>

### 基于人工鱼群算法的孪生支持向量机

Twin support vector machine based on artificial fish swarm algorithm

智能系统学报. 2019, 14(6): 1121–1126 <https://dx.doi.org/10.11992/tis.201905025>

### 改进猫群算法求解置换流水车间调度问题

Improved cat swarm optimization for permutation flow shop scheduling problem

智能系统学报. 2019, 14(4): 769–778 <https://dx.doi.org/10.11992/tis.201804016>

### 应用改进区块遗传算法求解置换流水车间调度问题

An improved puzzle-based genetic algorithm for solving permutation flow-shop scheduling problems

智能系统学报. 2019, 14(3): 541–550 <https://dx.doi.org/10.11992/tis.201801041>

### 求解离散优化问题的元胞量子狼群演化算法

Cellular and quantum-behaved wolf pack evolutionary algorithm for solving discrete optimization problems

智能系统学报. 2018, 13(5): 716–727 <https://dx.doi.org/10.11992/tis.201705007>

DOI: 10.11992/tis.202312042

网络出版地址: <https://link.cnki.net/urlid/23.1538.TP.20240909.1113.006>

# 基于多群体混合智能优化算法的卸载决策寻优方法

方浩添, 田乐, 郭茂祖

(北京建筑大学 建筑大数据智能处理方法研究北京市重点实验室, 北京 102600)

**摘要:** 在移动边缘计算的架构中, 为权衡降低计算应用卸载的能耗与时延, 引入卸载决策控制器, 并通过卸载决策寻优算法得到最优卸载决策。结合人工蜂群算法和人工鱼群算法提出新的人工蜂-鱼群 (artificial bee colony-fish swarm, ABC-FS) 算法, 在此基础上引入高斯衰减函数将算法参数由静态变为动态, 并将改进粒子群算法的惯性权重因子引入算法中, 从而得到一种多群体混合智能优化算法; 设计联合优化时延与能耗的目标函数, 再依据泊松概率进行仿真实验。仿真实验结果表明, 提出的卸载策略寻优算法, 与多组对照组相比, 收敛速度更快, 且在多接入边缘计算的场景下能权衡降低系统中任务卸载的总时延与总能耗。

**关键词:** 移动边缘计算; 计算卸载; 人工鱼群算法; 人工蜂群算法; 自相似排队模型; 高斯衰减函数; 粒子群算法; 惯性权重因子

中图分类号: TP393 文献标志码: A 文章编号: 1673-4785(2024)06-1573-11

中文引用格式: 方浩添, 田乐, 郭茂祖. 基于多群体混合智能优化算法的卸载决策寻优方法 [J]. 智能系统学报, 2024, 19(6): 1573-1583.

英文引用格式: FANG Haotian, TIAN Le, GUO Maozu. Unloading decision optimization method based on multi-population hybrid intelligent optimization algorithm[J]. CAAI transactions on intelligent systems, 2024, 19(6): 1573-1583.

## Unloading decision optimization method based on multi-population hybrid intelligent optimization algorithm

FANG Haotian, TIAN Le, GUO Maozu

(Beijing Key Laboratory of Intelligent Processing of Building Big Data, Beijing University of Civil Engineering and Architecture, Beijing 102600, China)

**Abstract:** In the network architecture of mobile edge computing, an offloading decision controller was introduced to balance the reduction of energy consumption and delay. This controller obtains the optimal offloading decision through an offloading decision optimization algorithm. A new ABC-FS algorithm was proposed by combining the artificial bee colony (ABC) algorithm and the artificial fish swarm (FS) algorithm. Additionally, a Gaussian decay function was introduced to transition the algorithm parameters from static to dynamic, and the inertia weight factor of the improved particle swarm optimization algorithm was incorporated, creating a multi-population hybrid intelligent optimization algorithm. Finally, an objective function that jointly optimizes delay and energy consumption was designed, and simulation experiments were conducted using Poisson probability. Simulation results show that the proposed offloading strategy optimization algorithm achieves faster convergence speed compared to several benchmark methods and effectively balances the reduction of total task offloading delay and total energy consumption in multi-access edge computing scenarios.

**Keywords:** moving edge computing; calculating offloading; artificial fish swarm algorithm; artificial colony algorithm; self-similar queuing model; gaussian attenuation function; particle swarm optimization; inertia weight factor

随着网络的不断发展, 云边缘架构中用户对终端计算能力的需求也不断增长, 当用户输入

计算密集型应用时, 终端设备往往无法在短时间内处理完成, 且高计算量的应用也会给终端设备带来高电量消耗, 为了解决这一问题, 欧洲电信标准协会提出了移动边缘计算的概念<sup>[1]</sup>。在移动边缘计算场景下, 边缘服务器和云服务器的计算

收稿日期: 2023-12-27. 网络出版日期: 2024-09-09.

基金项目: 国家自然科学基金项目 (62271036); 国家重点研发计划科技冬奥重点专项 (2021YFF0306303).

通信作者: 郭茂祖. E-mail: [guomaozu@bucea.edu.cn](mailto:guomaozu@bucea.edu.cn).

能力远高于终端设备,所以可以将终端设备上的计算任务传输到边缘服务器或云服务器进行计算<sup>[2]</sup>。

计算卸载是一种重要的技术策略,旨在通过将计算任务从一个设备转移至另一个拥有更强计算能力的设备来提高性能、效率并节能<sup>[3]</sup>。研究计算卸载可以提高移动设备能效、增强计算任务处理效率、优化计算资源分配、降低计算延迟。计算卸载的研究和应用对于推动计算技术的进步、提高用户体验以及促进新技术和服务的发展都有着重要的意义。

计算卸载决策寻优算法是为了在计算卸载的过程中做出最佳决策,即决定哪些计算任务应该在本地执行,哪些应该卸载到云端或边缘服务器,以达到特定的优化目标,如最小化延迟、能耗或成本。这些算法需要考虑多种因素,包括任务的计算需求、数据传输需求、网络条件、服务器的计算能力和成本等<sup>[4]</sup>。

本研究的目标是在云-边-端协同场景下,权衡终端设备上计算应用的时延与能耗,根据多群体混合智能优化算法以及设定的目标函数得到最优卸载决策,并根据最优卸载决策将计算任务发送到适合的节点上进行计算。首先,提出一个云边端协同移动边缘计算系统框架;然后,结合人工鱼群算法<sup>[5]</sup>、人工蜂群算法<sup>[6]</sup>、粒子群算法<sup>[7-8]</sup>以及高斯衰减函数提出了多群体混合智能优化算法;最后,通过仿真实验进行卸载决策寻优,与多组对照组进行对比,以验证本研究提出算法的性能。

## 1 相关工作

从近几年移动边缘计算的相关研究可知,计算卸载是研究热点。

目标最小化计算卸载时延的研究中,郝昊等<sup>[9]</sup>提出一种算网融合下时间连续的计算任务卸载机制,在保证时间轴连续和协同多个边缘服务器计算资源的前提下,以服务体验提升率为优化目标,对云、边、端间任务卸载问题进行建模,并设计了一种基于深度强化学习的任务卸载方法;姜玉龙等<sup>[10]</sup>针对链式工作流任务提出了一种基于势博弈的分布式工作流卸载算法,通过凸优化方法得到最优资源分配方案,将卸载问题建模成一个加权拥塞博弈,还针对有依赖性的工作流任务提出了一种基于动态资源权重的启发式工作流卸载算法,每个子任务基于资源权重选择完成时间最少的节点进行卸载。

目标为最小化计算卸载能耗的研究中,Zhang等<sup>[11]</sup>主要研究多用户在移动边缘计算系统中的

资源分配问题,通过利用回归问题解决通信与网络资源的分配问题,并通过合理分配网络通信与计算资源,在延迟约束下实现系统能源消耗最小化;Lyu等<sup>[12]</sup>设计了一个选择性卸载方案来最小化设备的能量消耗,通过允许设备自指定或自拒绝卸载,可以进一步降低信令开销。

目标为权衡计算卸载的时延与能耗的研究中,孙伟峰等<sup>[13]</sup>提出了一种新的基于双层 Stackelberg 博弈的计算卸载方案,将公有云作为私有云资源的补充,缓解了由于私有云资源局限性带来的算力不足问题;张祥俊等<sup>[14]</sup>在云边端协同计算卸载框架中引入 D2D (device-to-device) 通信和机会网络,将计算卸载决策问题转化为一个混合整数非线性规划问题,并对无线特性和移动用户之间的非合作博弈交互制定了一个迭代机制来共同确定计算卸载方案;邝祝芳等<sup>[15]</sup>通过深度强化学习提供了一个多用户多任务下的任务卸载调度管理和资源分配算法,在为上层系统分配资源确定的前提下,通过提供基于贪心策略的流水车间调度算法解决了多任务卸载管理和卸载调度的问题。

同时,也有很多研究将卸载策略求解转化为多因素优化问题,并运用群体智能算法求得最优卸载决策<sup>[16]</sup>。Feng等<sup>[17]</sup>结合灰狼优化算法和鲸鱼优化算法实现卸载决策的目标优化,并对比验证了混合算法的优越性;张文柱等<sup>[18]</sup>结合人工蜂群算法与粒子群优化算法来求解移动边缘计算中的卸载决策问题,提高了算法的寻优精度;王亭惠等<sup>[19]</sup>在人工鱼群算法的基础上加入狼群算法进行卸载决策寻优,提高了算法的收敛性;孔小爽等<sup>[20]</sup>结合鸟群算法和人工鱼群算法,提出一种区块链移动边缘计算卸载模型,将任务卸载问题转化为优化目标函数,通过降低任务时延和能量消耗来降低计算开销。

为了权衡降低计算卸载的时延与能耗,并加快计算卸载决策寻优的速度,本研究将多种群体智能算法结合,得到收敛速度快且寻优效率高的新算法,并依据新算法进行计算卸载决策寻优。

## 2 系统架构

系统架构由移动终端设备、卸载决策控制器、边缘服务器和云服务器组成,如图1所示。移动终端设备可以是智能手机、平板电脑、笔记本电脑以及无人机等设备;边缘服务器是位于网络边缘的服务器,用于存储和处理来自端设备的数据卸载;云服务器是部署在云中心的服务器;

卸载决策控制器部署在边缘服务器控制模块的一个控制节点上<sup>[21]</sup>。边缘服务器的选择方法是采用

生成随机数比大小的方式,选择数字最大的那台边缘服务器。

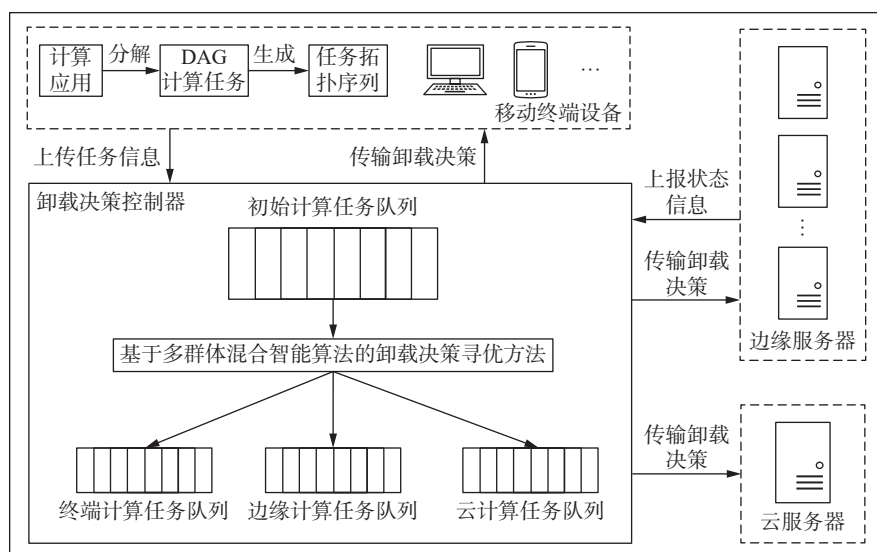


图1 系统框架

Fig. 1 System frame diagram

首先,用户在终端设备上输入计算应用,计算应用在终端设备上被分解成有依赖性的计算任务,通过任务的依赖关系画出有向无环图(directed acyclic graph, DAG),根据DAG图可以得到任务的拓扑序列。

然后,终端设备先把任务信息传输到卸载决策控制器上,所有终端设备上的任务拓扑序列在卸载决策控制器上集成成初始任务计算队列。同时,边缘服务器会向卸载决策控制器上报状态信息,在卸载决策控制器上通过多群体混合智能算法进行卸载决策的寻优,得到最终的云、边、端计算任务卸载队列。

最后,卸载决策控制器将计算任务卸载决策传回终端设备,终端设备根据卸载决策把任务发送到边缘节点和云节点,各节点进行计算,最后把计算结果传回终端设备。

### 3 多群体混合智能卸载决策寻优方法

#### 3.1 人工蜂-鱼群算法

人工蜂群算法存在易陷入局部最优的缺点<sup>[22]</sup>,人工鱼群算法能在可行解范围内全局搜索,克服了人工蜂群算法易陷入局部最优的问题;人工鱼群算法具有后期盲目寻优、收敛速度下降的缺点<sup>[23]</sup>,人工蜂群算法中3种不同的寻优角色在群体协作过程中有正、负反馈机制,能克服人工鱼群算法的问题。本研究结合人工蜂群算法的局部寻优能力和人工鱼群算法的全局寻优能力,提出收敛速度快且全局寻优能力强的人工蜂-鱼群算

法(artificial bee colony-fish swarm, ABC-FS),算法原理如图2所示。

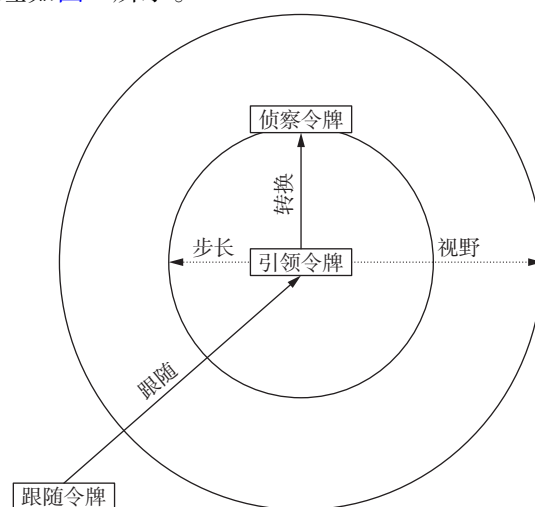


图2 ABC-FS 算法原理

Fig. 2 Schematic diagram of ABC-FS algorithm

算法中每个待卸载任务的卸载决策可以视为一个决策变量,表示为一个三态变量,0代表在本地执行,1代表卸载到边缘服务器,2代表卸载到云服务器。假设等待进行卸载的计算任务有 $N$ 个,一个解可以表示为一个包含 $N$ 个元素的向量,每个元素对应一个任务的卸载决策。例如,向量 $[0, 1, 1, 2, \dots]$ 表示第1个任务在本地执行,第2和第3个任务卸载到边缘服务器,第4个任务卸载到云服务器,依此类推。算法解空间由所有可能的卸载决策向量组成。

寻优令牌即为算法在解空间进行迭代寻优的个体,算法把寻优令牌个体分为3种角色,并为每



个令牌个体赋予移动步长与视野这2个参数。其中,引领令牌的作用是在算法每次迭代中寻找新的最优解,跟随令牌的作用是在算法每次迭代中保留引领令牌找到的最优解,侦察令牌的作用是使引领令牌跳出最优解。

假设等待进行卸载的计算任务有 $N$ 个,那么在算法解空间中需要设定 $2N$ 个令牌个体,其中 $N$ 个引领令牌, $N$ 个跟随令牌。算法解空间中的每一个寻优令牌个体均对应有一个待卸载任务,并具有一个 $N$ 维状态向量。

不同角色的行为和功能见表1。

表1 令牌各种类行为与功能的关系

Table 1 Relationship between the various class behaviors and functions of the token

令牌种类	行为	功能
引领令牌	搜寻最优解	保持优良解
跟随令牌	对最优解进行选择	加速算法收敛
侦察令牌	发现新的解	避免陷入局部最优

在随机生成每个令牌个体的初始状态向量后,寻优令牌可以在解空间进行寻优、聚集和追赶等行为,并不断迭代调整每个令牌个体的状态向量,直到最终确定目标函数值满足条件的对应状态向量。

ABC-FS 算法的流程如算法1所示。

#### 算法1 ABC-FS 算法

**输入** 输入与待卸载的计算任务数量相同的引领令牌和跟随令牌, $K$ 个终端设备, $M$ 个边缘节点, $N$ 个云节点。

**输出** 输出代表每个任务计算卸载地点的解 $X$ 。

- 1) **While** 当前迭代次数 $\leq$ 算法最大迭代次数
- 2) **for**  $i=1$ :待进行卸载的任务数
- 3) **if** 令牌群体中心位置不拥挤
- 4) 引领令牌进行聚集行为向中心位置移动;
- 5) **else**
- 6) 引领令牌进行寻优行为转换成目标函数值更优的新状态向量,寻优次数+1;
- 7) **if** 寻优次数超过最大寻优次数
- 8) 引领令牌转换为侦察令牌,然后进行随机行为,随机生成新的状态向量,再转换为新的引领令牌;
- 9) **else**
- 10) 引领令牌进行追赶行为和寻优行为转换成目标函数更优的新的状态向量,迭代次数+1;
- 11) 迭代次数+1;
- 12) **if** 迭代次数达到最大
- 13) 输出当前最优解;

14) **break**

15) **end**

### 3.2 算法参数调整

#### 3.2.1 感知范围和移动步长

如图2所示,在ABC-FS算法的寻优解空间中,每个寻优令牌个体都具有移动步长和感知范围这2个参数,在ABC-FS算法中只能固定地设定这2个参数。当移动步长和感知范围设定过小时,在算法前期的全局寻优阶段可能会错过最优解所在区域;当移动步长和感知范围设定过大时,在算法后期的局部寻优阶段可能会陷入局部最优解。

本研究把算法中移动步长和感知范围这2个由静态参数改为动态参数。在全局探索较优解时,利用较大的感知范围和移动步长充分开拓搜索空间,避免陷入局部最优解;在局部小范围搜索时,缩短步长及感知范围,加强搜索能力,提高寻优精度。

具体的方法是,结合算法迭代次数,为各参数引入一个权值,通过控制这个权值,使得在迭代初期,解空间中的寻优令牌能够以较大的感知范围和步长移动,加快收敛;而在迭代中后期令牌以较小的感知范围和步长逼近极值,以取得较高的优化精度。

#### 3.2.2 衰减函数的引入

本研究选择在静态参数中引入衰减函数,从而通过控制衰减函数而使静态参数变为动态参数。

张毅等<sup>[24]</sup>在人工鱼群算法中引入指数衰减函数对算法参数进行控制,本研究选择在ABC-FS算法中引入比指数函数更优的高斯衰减函数。

高斯衰减函数依据高斯分布,即正态分布。依据正态分布的性质以及高斯衰减函数公式,设算法当前迭代次数为 $t$ ,算法最大迭代次数为 $T$ ,只取高斯函数一半的曲线,位置参数 $\mu$ 取1,依据正态分布 $3\sigma$ 原则, $3\sigma=T$ ,那么 $\sigma=T/3$ ,根据高斯衰减公式,可得出:

$$\text{gauss}(t) = \exp \left( -\frac{(t-1)^2}{2 \times \left( \frac{T}{3} - 1 \right)^2} \right)$$

将上述高斯衰减函数与张毅等<sup>[24]</sup>研究中提出的指数衰减函数进行对比,如图3所示。

对于ABC-FS算法解空间中寻优令牌个体的感知范围和移动步长,如果感知范围和步长在迭代的前期能缓慢衰减,保持较大的值,就能保证前期的全局收敛速度;在迭代的中期感知范围和

步长快速衰减,到后期衰减放缓,能提高寻优令牌的局部搜索能力,在最优邻域内进行精确搜寻。

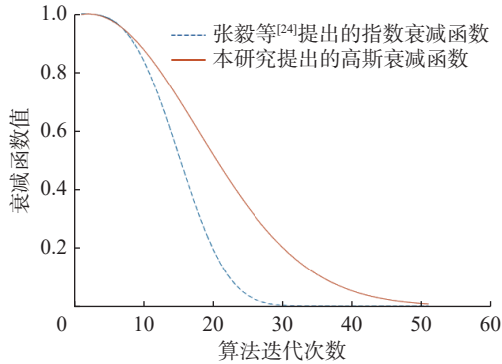


图3 衰减函数对比

Fig. 3 Attenuation function comparison diagram

依据图3,由衰减曲线斜率变化可明显看出,高斯衰减函数在算法迭代前期衰减速度小于指数衰减函数,而在迭代中后期高斯衰减函数衰减速度明显高于指数衰减函数,综上得出高斯衰减函数优于指数衰减函数。

### 3.2.3 衰减函数与参数的结合

本研究利用高斯衰减函数将ABC-FS算法中的感知范围与移动步长这2个参数从静态变为动态,首先设定移动步长step与感知范围visual的最大值与最小值,然后结合高斯衰减函数(gauss( $t$ )),最后运用max函数分别得到:

$$\text{step} = \max \{ \text{step}_{\max} \times \text{gauss}(t), \text{step}_{\min} \}$$

$$\text{visual} = \max \{ \text{visual}_{\max} \times \text{gauss}(t), \text{visual}_{\min} \}$$

如果参数最小值设定过大,那么参数在算法迭代前期就会完成衰减过程,减缓了算法的收敛速度,也降低了算法的寻优精度;如果参数最小值设定过小,那么在算法迭代后期容易陷入局部最优。经过多次实验论证,最终选择将移动步长和感知范围的最小值都设定成为初始值的2/5,可得到2个参数的衰减变化,如图4所示。

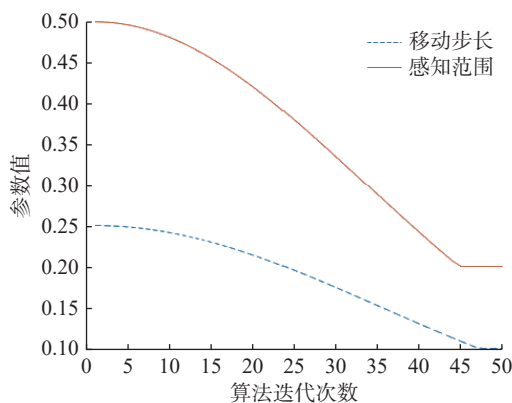


图4 参数衰减变化

Fig. 4 Parameter attenuation variation diagram

由图4可以看出,参数在迭代前期缓慢衰减,在迭代中后期迅速衰减,且衰减到先前设定的最小值就结束衰减,符合预期。

### 3.2.4 动态惯性权重因子

在传统的粒子群算法中,当惯性权重因子较大时,寻优个体的全局寻优能力强;当附加的惯性权重因子较小时,寻优个体的局部寻优能力强。如果在算法迭代中一直使用较小的惯性权重因子,会使算法陷入局部最优解[25]。

在ABC-FS算法中,解空间中的寻优令牌个体的寻优行为具有随机性,并存在一定的局限性;聚集行为可以帮助寻优令牌找到自身的中心位置,但容易得到局部最优;追赶行为可以找到视野内最优解,但行为条件比较单一;且ABC-FS算法还存在后期收敛速度慢,易陷入局部最优的不足。

为了提高ABC-FS算法迭代前期在解空间中的全局寻优能力以及算法迭代中后期在解空间中的局部寻优能力,本研究提出动态惯性权重因子的概念。

将惯性权重因子的最大值设定成0.9,最小值设定成0.4,并带入之前求得的高斯衰减函数,得到本研究新提出的动态惯性权重因子:

$$\omega = \omega_{\min} + (\omega_{\max} - \omega_{\min}) \times \text{gauss}(t)$$

惯性权重因子根据算法迭代次数变化的图像如图5所示,由图5可以看出,由于高斯衰减函数的加入,惯性权重因子具有衰减趋势,且衰减速度随着算法迭代次数不断减小。

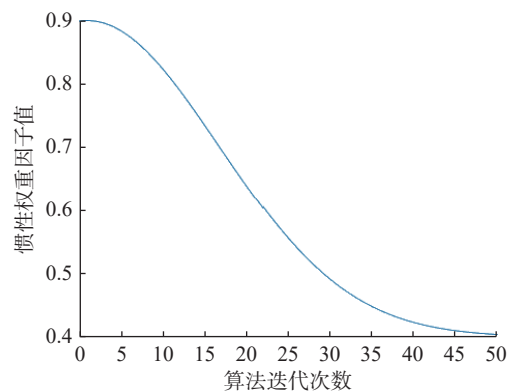


图5 动态惯性权重因子变化

Fig. 5 Dynamic inertia weight factor variation diagram

### 3.2.5 更改寻优个体移动公式

原ABC-FS算法中,寻优个体的移动公式为

$$\text{移动方向} = X_1 + \text{rand}() \times \text{step} \times \frac{X_2 - X_1}{\|X_2 - X_1\|}$$

式中: $X_1$ 为当前位置, $X_2$ 为目标位置,rand()为0~1的随机数,step为寻优令牌个体在解空间中的最大移动步长。

将上文所提出的新的惯性权重因子加入ABC-FS算法,随着算法迭代次数不断增大,惯性权重因子由大到小动态变化,相当于在解空间的每一次迭代中,为寻优令牌个体的目标位置附加一个动态权值。

在算法迭代前期,为了加强算法的全局寻优能力,避免算法陷入局部最优,使用较大的惯性权重因子;在算法迭代中后期,为了加强算法的局部寻优能力,加快算法收敛速度,使用较小的惯性权重因子。

新的寻优个体的移动公式为

$$\text{移动方向} = X_1 + \omega \times \text{rand}() \times \text{step} \times \frac{X_2 - X_1}{\|X_2 - X_1\|}$$

式中  $\omega$  为惯性权重因子。

### 3.3 理论分析

#### 3.3.1 计算模型参数

计算模型中的参数解释说明见表2。

表2 计算模型中的参数  
Table 2 Calculate the parameters in the model

参数名	符号
流量方差系数	$a$
自相似系数	$H$
计算任务平均到达速率	$m$
伽马函数	$\Gamma(x)$
任务的平均服务速率	$\mu$
任务 $k$ 的计算量( $\text{amo}_k$ )	$a_{\text{mok}}$
信道带宽( $\text{ban}$ )	$b_{\text{an}}$
信道噪声功率( $\text{noi}^2$ )	$n_{\text{oi}}^2$
链路之间干扰功率( $\text{int}_n$ )	$i_{\text{ntn}}$
终端设备计算能力( $\text{abi}_m$ )	$a_{\text{bim}}$
终端设备可用CPU资源( $\text{cpu}_m$ )	$c_{\text{pum}}$
终端设备可用内存资源( $\text{mem}_m$ )	$m_{\text{emm}}$
终端设备发射功率( $\text{pow}_m$ )	$p_{\text{owm}}$
终端设备由芯片结构决定的开关电容( $\text{cap1}$ )	$c_{\text{ap1}}$
边缘服务器由芯片结构决定的开关电容( $\text{cap2}$ )	$c_{\text{ap2}}$
云服务器由芯片结构决定的开关电容( $\text{cap3}$ )	$c_{\text{ap3}}$
边缘服务器 $e$ 的最大计算能力( $\text{abi}_e$ )	$a_{\text{bie}}$
边缘服务器可用CPU资源( $\text{cpu}_e$ )	$c_{\text{pue}}$
边缘服务器可用内存资源( $\text{mem}_e$ )	$m_{\text{eme}}$
边缘服务器与终端设备间传输速度( $\text{spe}_e$ )	$s_{\text{pee}}$
边缘服务器发射功率( $\text{pow}_m$ )	$p_{\text{owm}}$
云服务器最大计算能力( $\text{abi}_c$ )	$a_{\text{bic}}$
云服务器与终端设备间传输速度( $\text{spe}_c$ )	$s_{\text{pec}}$
云服务器发射功率( $\text{pow}_c$ )	$p_{\text{owc}}$
任务计算的结果数据( $\text{data}_{re}$ )	$d_{\text{atare}}$
任务的本地数据( $\text{data}_k$ )	$d_{\text{atak}}$

#### 3.3.2 边缘节点排队时延模型

本研究利用自相似性性质建立计算任务的排队模型并进行时延分析。近年来对计算卸载的研究表明,在计算任务卸载到边缘服务器时,大多数情况下是多个计算任务连续到达,这意味着计算任务到达的过程具有突发性,即计算任务卸载到边缘服务器的过程具有自相似性<sup>[26]</sup>。

分形布朗运动(fractal Brown motion, FBM)是一种描述时间序列自相似性的非平稳随机过程,FBM模型是能够描述网络业务流自相似特性的经典的自相似网络流量模型,依据FBM的性质,计算任务到达边缘服务器的过程服从分形布朗运动<sup>[27]</sup>。已知经过卸载调度的计算任务到达边缘服务器的时间间隔序列  $X_i = \{X_1, X_2, \dots, X_n\}$ , 首先计算出  $t$  的均值  $x_n$ 、标准差  $S$ 、流量方差系数  $a$ :

$$x_n = \frac{1}{n} \sum_{i=1}^n X_i$$

$$S = \sqrt{\frac{1}{n} \sum_{i=1}^n (X_i - x_n)^2}$$

$$a = \frac{S}{x_n}$$

然后使用重标极差分析法计算出自相似参数Hurst的值,采用无量纲比值  $R/S$  (极差/方差)对极差进行重新标度,首先定义序列  $W_j$ :

$$W_j = (X_1 + X_2 + \dots + X_j) - jx_n, j = 1, 2, \dots, n$$

则极差( $R$ )为

$$R = \max(0, W_1, W_2, \dots, W_n) - \min(0, W_1, W_2, \dots, W_n)$$

将极差与方差画在双对数坐标轴上,对该曲线进行直线拟合,直线斜率即为Hurst参数。

假设终端设备卸载的计算任务到达边缘服务器的时间间隔序列相互独立,且服从FBM模型,可推导出平均队列长度  $E(B)$ :

$$E(B) = \left( \frac{2amH^{2H} (1-H)^{2-2H}}{(\mu-m)^{2H}} \right)^{\frac{1}{2-2H}} \times \Gamma\left(\frac{1}{2-2H} + 1\right)$$

式中  $\mu$  为任务的平均服务速率。

根据李特尔公式可以推导出边缘服务器中的平均排队时延  $T_d$ :

$$T_d = \frac{E(B)}{\mu}$$

#### 3.3.3 云边端时延能耗模型

任务由终端设备本地进行运算的时延  $T_M$  和能耗  $E_M$ :

$$T_M = \frac{a_{\text{mok}}}{a_{\text{bim}}}$$

$$E_M = c_{\text{ap1}} \times a_{\text{bim}}^2 \times a_{\text{mok}}$$

本系统中  $e$  个边缘服务器的集合为  $E_{\text{dge}} = \{E_{d_1}, E_{d_2}, \dots, E_{d_e}\}$ , 每个边缘服务器的基本属性为



$$E_{de} = \{a_{bie}, c_{pue}, m_{eme}, s_{pee}, p_{owe}\}$$

假设终端设备与边缘服务器之间有  $n$  条链路, 并且每条链路只允许被一台终端设备占用。由 Shannon 公式的边缘服务器与移动设备间的数据传输速度为

$$s_{pee} = b_{an} \times \log_2 \left( 1 + \frac{p_{owm}}{n_{oi}^2 + i_{nbi}} \right)$$

任务  $k$  到达边缘服务器  $e$  后, 若边缘服务器当前剩余资源满足任务计算所用资源, 则任务  $k$  可立即执行, 其总完成时间为

$$T_k = \frac{d_{atak}}{s_{pee}} + \frac{a_{mok}}{a_{bie}} + \frac{d_{atare}}{s_{pee}}$$

根据上文得到的 FBM 自相似排队模型的排队时延  $T_d$ , 可得到任务  $k$  卸载到边缘服务器  $e$  的总时延:

$$T_{ED} = T_k + T_d$$

任务在边缘服务器进行计算的能耗  $E_{ED}$  为

$$E_{ED} = p_{owm} \times \frac{d_{atak}}{s_{pee}} + c_{ap2} \times a_{bie}^2 \times a_{mok} + p_{owm} \times \frac{d_{atare}}{s_{pee}}$$

本研究中的系统部署一个云服务器, 因为云服务器的内存资源和计算资源远高于终端设备和边缘服务器所拥有的资源, 所以只考虑云服务器的 3 个属性:

$$C = \{a_{bic}, s_{pec}, p_{owc}\}$$

任务卸载到云中心的时延  $T_C$  为

$$T_C = \frac{d_{atak}}{s_{pec}} + \frac{a_{mok}}{f_c} + \frac{d_{atare}}{s_{pec}}$$

云中心的能耗为

$$E_{ED} = p_{owm} \times \frac{d_{atak}}{s_{pec}} + c_{ap3} \times a_{bic}^2 \times a_{mok} + p_{owm} \times \frac{d_{atare}}{s_{pec}}$$

### 3.3.4 目标函数的设定

卸载决策的总时延  $T_{all}$  和总能耗  $E_{all}$  分别为

$$T_{all} = \sum_{i=1}^n T(i)$$

$$E_{all} = \sum_{i=1}^n E(i)$$

在移动边缘计算环境下的计算卸载决策寻优中, 为了权衡降低任务计算的时间和能耗, 本研究在张文柱等<sup>[18]</sup>研究的目标函数中引入边缘侧计算的指标, 设计了科学评价任务卸载效果的目标函数  $Q$ :

$$Q = \frac{1}{2} \times \left( 1 - \frac{T_{all}}{\frac{1}{3} \times (T_{allM} + T_{allED} + T_{allC})} \right) + \frac{1}{2} \times \left( 1 - \frac{E_{all}}{\frac{1}{3} \times (E_{allM} + E_{allED} + E_{allC})} \right)$$

同时考虑时延和能耗两方面因素, 将 2 个不同的优化变量整合到一个目标函数中, 通过求目标函数值的最大值, 得到时延和能耗权衡最小的最优卸载决策。

任务全部在终端设备本地执行时的时延和能耗由  $T_{allM}$  和  $E_{allM}$  表示, 任务全部在边缘服务器执行时的时延和能耗由  $T_{allED}$  和  $E_{allED}$  表示, 任务全部在云服务器执行时的时延和能耗由  $T_{allC}$  和  $E_{allC}$  表示。将以上参数相加再除以 3 表示基准值, 算取时延和能耗的标么值, 再归一化, 得到目标函数  $Q$ 。 $Q$  越大说明卸载决策中时延和能耗权衡最小。

## 4 仿真实验

### 4.1 数据集的选择与处理

在真实的计算卸载环境中, 用户在终端设备输入计算应用, 输入的应用在终端设备本地被分解成具有依赖性的 DAG 任务, 如图 6 所示。



图 6 计算应用分解过程

Fig. 6 Computationally applied decomposition process

为了模仿真实的计算卸载, 本研究通过阿里巴巴公司 Cluster Trace 数据集获取任务 DAG 结构, 该数据集基于阿里巴巴公司开源项目开放式集群跟踪计划 (alibaba cluster trace program)<sup>[28]</sup>, 数据集记录了某个包含大约 4 000 台机器的生产集群中服务器以及运行任务在 8 d 内的执行信息, 也记录了提交到阿里云上 4 000 万个批处理作业的详细信息, 并以“作业-任务-实例”方式刻画批处理作业, 其中包括了生产负载中的 DAG 依赖结构信息, 可以对批处理作业中的原数据进行分析, 提取出相关的 DAG 结构。

数据集内的 DAG 结构一共有 3 种, 即无依赖关系、简单依赖关系以及复杂依赖关系, 每一个 DAG 结构模拟一个由用户输入终端设备的计算应用, 每个应用被分解成有依赖性的任务, 根据 DAG 结构得到任务的拓扑序列, 再整合得到待卸载的初始任务计算队列。

### 4.2 仿真实验介绍

本研究使用仿真工具, 在 Windows 系统下对提出的算法进行实验验证, 证明算法的有效性, 实验环境见表 3。



表3 仿真实验环境

Table 3 Simulation experiment environment

系统版本	处理器	内存大小	硬盘容量
Windows 10 专业版	11th Gen Intel(R) Core(TM) i7-11700 @ 2.50 GHz 2.50 GHz	16.0 GB	1 TB

仿真实验中使用的相关参数参考 Zhou 等<sup>[29]</sup> 的设置, 见表4。

表4 实验参数设定

Table 4 Experimental parameter setting

参数	参数值
移动终端设备数量 $M$	25
边缘服务器数量 $E$	4
信道带宽 $b_{an}/\text{MHz}$	20
信道噪声功率 $n_{oi}^2/\text{W}$	$2 \times 10^{-13}$
链路之间干扰功率 $i_{intr}/\text{W}$	$2 \times 10^{-13}$
终端设备计算能力 $a_{bim}/\text{GHz}$	4
终端设备发射功率 $p_{own}/\text{W}$	[0.1, 0.5]
开关电容 $c_{ap}/\mu\text{F}$	0.01
边缘服务器最大计算能力 $a_{bic}/\text{GHz}$	50
边缘服务器可用CPU资源 $c_{pue}/\text{GHz}$	[3.0, 30.0]
边缘服务器可用内存资源 $m_{eme}/\text{MB}$	[1, 40]
边缘服务器与终端设备间传输速度 $s_{pee}/(\text{Mbit/s})$	[1, 10]
边缘服务器发射功率 $p_{own}/\text{W}$	4
云服务器最大计算能力 $a_{bic}/\text{GHz}$	100
云服务器与终端设备间传输速度 $s_{pec}/(\text{Mbit/s})$	[1, 10]
云服务器发射功率 $p_{owc}/\text{W}$	10
任务到达速率 $m/(\text{bit/s})$	60
自相似系数 $H$	0.75
流量方差系数 $a$	30
服务速率 $\mu/(\text{bit/s})$	1 000

为了验证 ABC-FS 算法的优越性, 将通过算法寻得的最优卸载决策与 6 种任务处理方式做比较见表5。

表5 实验对照组

Table 5 Experimental control group

1	2	3	4	5	6
任务只 在终端 设备进 行处理 计算	任务全部 卸载到边 缘计算卸 载到边缘 计算	任务全部 卸载到云 中心计算	任务通过人 工蜂群算法 得到最优卸 载决策进行 计算	任务通过人 工鱼群算法 得到最优卸 载决策进行 计算	任务通 过ABC- FS算法 得到最 优卸载 决策进 行计算

### 4.3 终端设备任务的泊松分布

在阿里数据集中选定属于 23 个 DAG 图的 50 个计算任务, 模拟 23 个用户输入终端设备的初始计算应用。首先给 25 个终端设备标号, 由图7的泊松概率可知, 在 1~25 个终端设备中, 只有第 9~23 号终端设备有概率被用户输入计算应用。

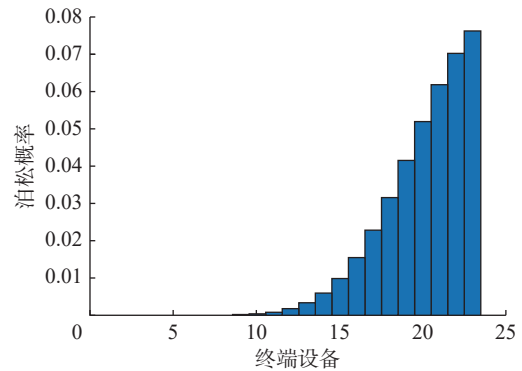


图7 任务分布的泊松概率

Fig. 7 Poisson probability of task distribution

为了模拟真实的用户输入计算应用的情况, 将 23 个计算应用按照泊松概率分配到 15~23 号终端设备上, 即 15~17 号终端设备各分配 1 个计算应用, 18 号终端设备分配 2 个计算应用, 19 号和 20 号终端设备各分配 3 个计算应用, 21~23 号终端设备各分配 4 个计算应用。

在接下来的研究中, 将多次泊松概率分布的计算任务分配结果分别进行实验仿真, 实验结果取平均值。

### 4.4 实验结果分析

算法运行过程中目标函数值  $Q$  的变化如图8所示, 随着算法的不断迭代,  $Q$  值逐渐趋于平稳, 即达到最优值。

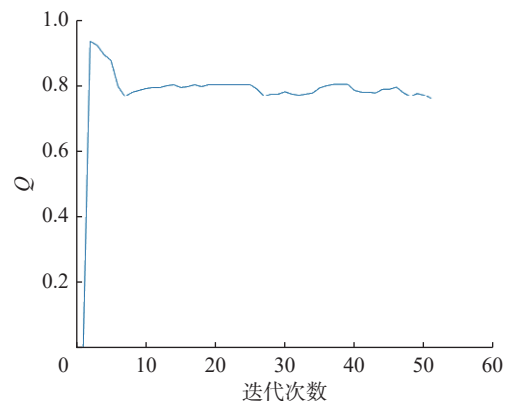


图8 目标函数值的迭代变化

Fig. 8 Iterative changes in the value of the objective function

根据 5 次不同的泊松概率分布在终端设备上分配任务, 进行 5 次仿真实验, 结果取平均值。将改进后的 ABC-FS 算法得到的结果与 6 个不同的

实验对照组进行对比。

本研究的寻优目标是找到计算卸载时延与能耗均衡最小的卸载决策,由图9和图10可知,只在终端计算和根据人工蜂群算法卸载的能耗都较小,但是时延较大;只在边缘服务器计算与只在云服务器计算的时延都较小,但是能耗较大;根据人工鱼群算法卸载、根据ABC-FS算法卸载以及根据多群体混合智能优化算法卸载的时延和能耗都较均衡,但是明显多群体混合智能优化算法的结果更优。

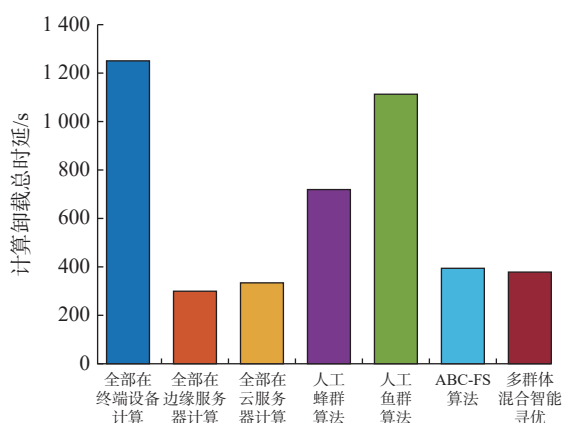


图9 卸载时延对比

Fig. 9 Unload delay comparison

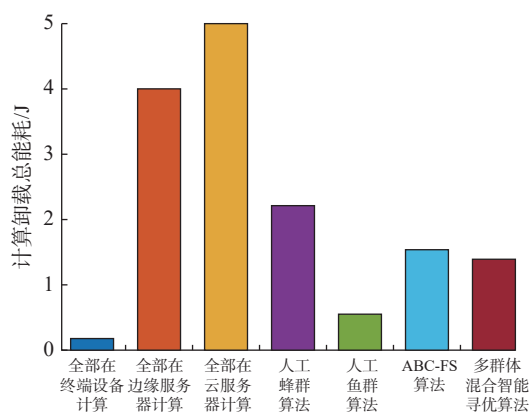


图10 卸载能耗对比

Fig. 10 Comparison of discharge energy consumption

由图11可知,将根据人工鱼群算法、人工蜂群算法、改进前的ABC-FS算法以及多群体混合智能寻优算法寻优结果计算的目标函数值 $Q$ 进行对比,可以得出多群体混合智能优化算法的目标函数值更优。由图12可知,因为本研究改进ABC-FS算法的目的是缩短算法收敛时间,将ABC-FS算法与多群体混合智能优化算法的运行时间采用柱状图进行对比,可明显看出多群体混合智能优化算法运行时间变短,即算法的收敛速度变快。

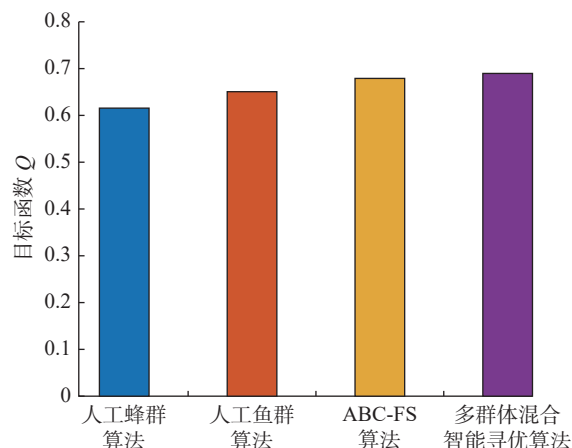


图11 不同算法 $Q$ 对比

Fig. 11 Comparison of  $Q$  values of different algorithms

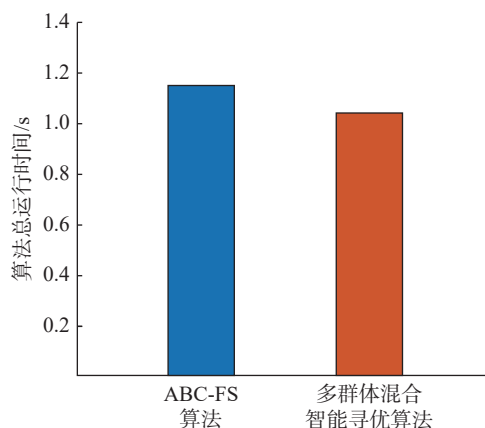


图12 改进前后算法运行时间对比

Fig. 12 Comparison of algorithm running time before and after improvement

## 5 结束语

本研究融合了人工鱼群算法与人工蜂群算法的优点,提出一种基于ABC-FS算法的移动边缘计算卸载模型,结合高斯衰减函数与粒子群算法的惯性权重因子,得到一种多群体混合智能优化算法,并设计了联合优化时延与能耗的目标函数。仿真结果表明,本研究算法具有更好的收敛性,能够有效权衡降低系统总能耗和计算卸载时延。未来本研究计划结合人工智能与机器学习等技术,对多群体混合智能优化算法进一步改进。

## 参考文献:

- [1] MACH P, BECVAR Z. Mobile edge computing: a survey on architecture and computation offloading[J]. IEEE communications surveys & tutorials, 2017, 19(3): 1628-1656.
- [2] 薛建强, 史彦军, 李波. 面向无人集群的边缘计算技术综述[J]. 兵工学报, 2023, 44(9): 2546-2555.  
XUE Jianqiang, SHI Yanjun, LI Bo. A review of edge

- computing technology for unmanned swarms[J]. *Acta armamentarii*, 2023, 44(9): 2546–2555.
- [3] 张冰洁, 杨彦红, 曹少中. 面向多接入边缘计算的计算卸载方案研究综述[J]. *计算机科学与探索*, 2023, 17(9): 2030–2046.
- ZHANG Bingjie, YANG Yanhong, CAO Shaozhong. Review of computing offloading schemes for multi-access edge computing[J]. *Journal of frontiers of computer science and technology*, 2023, 17(9): 2030–2046.
- [4] 孔珊, 柴争义, 郑玉琦. 基于进化多任务多目标优化的边缘计算任务卸载[J]. *计算机应用研究*, 2024, 41(4): 1164–1170.
- KONG Shan, CHAI Zhengyi, ZHENG Yuqi. Task offload in edge computing based on evolutionary multi task multi-objective optimization[J]. *Application research of computers*, 2024, 41(4): 1164–1170.
- [5] KARABOGA D. An idea based on honey bee swarm for numerical optimization[R]. Technical report TR06, Erciyes University, 2005(TR06).
- [6] 李晓磊. 一种新型的智能优化方法—人工鱼群算法[D]. 杭州: 浙江大学, 2003.
- LI Xiaolei. A new intelligent optimization Method-Artificial fish school algorithm[D]. Hangzhou: Zhejiang University, 2003.
- [7] EBERHART R, KENNEDY J. A new optimizer using particle swarm theory[C]//MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science. Nagoy: IEEE, 2002: 39–43.
- [8] KENNEDY J, EBERHART R. Particle swarm optimization[C]//Proceedings of ICNN'95 - International Conference on Neural Networks. Perth: IEEE, 2002: 1942–1948.
- [9] 郝昊, 杨树杰, 张玮. 算网融合下时间连续的计算任务卸载机制[J]. *计算机研究与发展*, 2023, 60(4): 735–749.
- HAO Hao, YANG Shujie, ZHANG Wei. Time-continuous computing task offloading mechanism for computing and network convergence[J]. *Journal of computer research and development*, 2023, 60(4): 735–749.
- [10] 姜玉龙, 东方, 郭晓琳, 等. 算力网络环境下基于势博弈的工作流任务卸载优化机制[J]. *计算机研究与发展*, 2023, 60(4): 797–809.
- JIANG Yulong, DONG Fang, GUO Xiaolin, et al. Potential game based workflow task offloading optimization mechanism in computing power network[J]. *Journal of computer research and development*, 2023, 60(4): 797–809.
- [11] ZHANG Y, ZHOU X, TENG Y, et al. Resource allocation for multi-user mobile edge computing system: machine learning approaches[C]//Proceedings of the 2018 International Conference on Computational Science and Computational Intelligence. Las Vegas: IEEE, 2018: 794–799.
- [12] LYU Xinchun, TIAN Hui, JIANG Li, et al. Selective offloading in mobile edge computing for the green Internet of Things[J]. *IEEE network*, 2018, 32(1): 54–60.
- [13] 孙伟峰, 张渊樾, 江贺, 等. 基于双层 Stackelberg 博弈的 MEC 计算卸载方案[J]. *软件学报*, 2023, 34(9): 4275–4293.
- SUN Weifeng, ZHANG Yuankui, JIAN He, et al. Computation offloading scheme based on two-layer stackelberg game for multi-access edge computing[J]. *Journal of software*, 2023, 34(9): 4275–4293.
- [14] 张祥俊, 伍卫国, 张弛, 等. 面向移动边缘计算网络的高能效计算卸载算法[J]. *软件学报*, 2023, 34(2): 849–867.
- ZHANG Xiangjun, WU Weiguo, ZHANG Chi, et al. Energy-efficient computing offloading algorithm for mobile edge computing network[J]. *Journal of software*, 2023, 34(2): 849–867.
- [15] 邝祝芳, 陈清林, 李林峰, 等. 基于深度强化学习的多用户边缘计算任务卸载调度与资源分配算法[J]. *计算机学报*, 2022, 45(4): 812–824.
- KUANG Zhufang, CHEN Qinglin, LI Linfeng, et al. Multi-user edge computing task offloading scheduling and resource allocation based on deep reinforcement learning[J]. *Chinese journal of computers*, 2022, 45(4): 812–824.
- [16] KAR B, YAHYA W, LIN Y D, et al. A survey on offloading in federated cloud-edge-fog systems with traditional optimization and machine learning[EB/OL]. (2022–02–22)[2022–12–12]. <http://arxiv.org/abs/2202.10628>.
- [17] FENG Siling, CHEN Yinjie, ZHAI Qianhao, et al. Optimizing computation offloading strategy in mobile edge computing based on swarm intelligence algorithms[J]. *EURASIP journal on advances in signal processing*, 2021, 2021(1): 36.
- [18] 张文柱, 余静华. 移动边缘计算中基于云边端协同的任务卸载策略[J]. *计算机研究与发展*, 2023, 60(2): 371–385.
- ZHANG Wenzhu, YU Jinghua. Task offloading strategy in mobile edge computing based on cloud-edge-end co-operation[J]. *Journal of computer research and development*, 2023, 60(2): 371–385.
- [19] 王亭惠, 陈桂芬. 基于 DP-HAFS 人工鱼群算法的移动边缘计算卸载策略[J]. *计算机应用研究*, 2023, 40(4): 1184–1188.
- WANG Tinghui, CHEN Guifen. Mobile edge computing offloading strategy based on DP-HAFS algorithm[J]. *Application research of computers*, 2023, 40(4): 1184–1188.
- [20] 孔小爽, 袁健. 基于鸟群人工鱼群算法的区块链移动边缘计算卸载模型[J]. *电子科技*, 2024, 37(8): 26–33.



- KONG Xiaoshuang, YUAN Jian. Blockchain mobile edge computing offloading model based on bird swarm artificial fish swarm algorithm[J]. Electronic science and technology, 2024, 37(8): 26–33.
- [21] 齐小刚, 单明媚, 张皓然. 软件定义网络故障诊断综述[J]. 智能系统学报, 2023, 18(4): 662–675.
- QI Xiaogang, SHAN Mingmei, ZHANG Haoran. Summary of software-defined networking fault diagnosis[J]. CAAI transactions on intelligent systems, 2023, 18(4): 662–675.
- [22] 章呈瑞, 柯鹏, 尹梅. 改进人工蜂群算法及其在边缘计算卸载的应用[J]. 计算机工程与应用, 2022, 58(7): 150–161.
- ZHANG Chengrui, KE Peng, YIN Mei. Improved artificial bee colony algorithm and its application in edge computing offloading[J]. Computer engineering and applications, 2022, 58(7): 150–161.
- [23] 安家乐, 刘晓楠, 何明, 等. 量子群智能优化算法综述[J]. 计算机工程与应用, 2022, 58(7): 31–42.
- AN Jiale, LIU Xiaonan, HE Ming, et al. Survey of quantum swarm intelligence optimization algorithm[J]. Computer engineering and applications, 2022, 58(7): 31–42.
- [24] 张毅, 杨光辉, 花远红. 基于改进人工鱼群算法的机器人路径规划[J]. 控制工程, 2020, 27(7): 1157–1163.
- ZHANG Yi, YANG Guanghui, HUA Yuanhong. Robot path planning based on improved artificial fish swarm algorithm[J]. Control engineering of China, 2020, 27(7): 1157–1163.
- [25] 王生亮, 刘根友. 一种非线性动态自适应惯性权重 PSO 算法[J]. 计算机仿真, 2021, 38(4): 249–253, 451.
- WANG Shengliang, LIU Genyou. A nonlinear dynamic adaptive inertial weight particle swarm optimization [J]. Computer integrated manufacturing systems, 2021, 38(4): 249–253, 451.
- [26] 魏德宾, 潘成胜, 杨力, 等. 基于自相似网络流量预测的自适应随机早期检测算法[J/OL]. 通信学报: 1–13 [2023–07–05]. <http://kns.cnki.net/kcms/detail/11.2102.tn.20230625.1700.010.html>.
- WEI Debin, PAN Chengsheng, YANG Li, et al. Adaptive random early detection algorithm based on self-similar network traffic prediction[J/OL]. Journal on Communications: 1–13 [2023–07–05]. <http://kns.cnki.net/kcms/detail/11.2102.tn.20230625.1700.010.html>.
- [27] 弭娜. SDN 控制器放置及扩容性能优化验证平台[D]. 济南: 山东大学, 2021.
- MI Na. Validation and verification platform of sdn controller placement and incremental capacity optimization [D]. Jinan: Shandong University, 2021.
- [28] Alibaba Cluster Trace Program. Alibaba production cluster data [EB/OL]. [2022–01–01]. <https://github.com/alibaba/clusterdata/tree/v2018>.
- [29] ZHOU Ping, SHEN Ke, KUMAR N, et al. Communication-efficient offloading for mobile-edge computing in 5G heterogeneous networks[J]. IEEE internet of things journal, 2021, 8(13): 10237–10247.

### 作者简介:



方浩添, 硕士研究生, 主要研究方向为移动边缘计算。E-mail: [2108110021005@stu.bucea.edu.cn](mailto:2108110021005@stu.bucea.edu.cn)。



田乐, 副教授, 博士, 主要研究方向为计算机网络、无线通信、大数据处理。E-mail: [tianle@bucea.edu.cn](mailto:tianle@bucea.edu.cn)。



郭茂祖, 教授, 博士生导师, 主要研究方向为机器学习、数据挖掘理论与算法、智能建造与智慧城市等人工智能应用、生物信息学。获得省杰出青年科学基金项目(2006年), 教育部宝钢优秀教师奖(2015年)等荣誉称号, 教育部自然科学奖(2019年)、第十届吴文俊人工智能自然科学奖(2020年)以及其他省部级科技奖4项(2002、2008年)、省部级教学成果奖1项(2018年)。主持、参与过3项自然科学基金委重点项目等, 发表学术论文290余篇, 授权专利、软件著作权20余项。E-mail: [guomaozu@bucea.edu.cn](mailto:guomaozu@bucea.edu.cn)。