

边缘计算中基于QoE感知的任务卸载:势博弈方法

师子琦¹, 谢刚^{1*}, 范文浩², 刘元安²

(1. 北京邮电大学信息与通信工程学院, 北京 100876; 2. 北京邮电大学电子工程学院, 北京 100876)

摘要: 随着5G的广泛应用,边缘计算技术被用于任务卸载和处理,基于博弈论的边缘计算策略成为当前研究领域的热点. 本文以最大化用户体验质量(Quality of Experience, QoE)为目标,研究时间约束条件下的多用户任务卸载问题. 本文首先从通信模型、计算模型和时间约束三个方面建立系统模型,然后将优化问题转换为博弈问题,给出并证明存在纳什均衡解. 本文提出了一种分布式多用户卸载算法(Distributed Multi-User Offloading, DMUO),首次实现了多用户在单时隙内同步更新策略,显著降低了计算开销并提升了收敛速度. 理论分析表明,DMUO算法能够收敛至纳什均衡解,并给出了迭代次数的上限. 此外,通过分析最坏情况策略与最优解的性能差距,验证了算法的鲁棒性. 仿真实验表明,DMUO算法具有优异的收敛性和系统性能,证明了其在大规模边缘计算环境中的可扩展性和实际适用性.

关键词: 移动边缘计算; 势博弈; 纳什均衡; 计算卸载; 体验质量

基金项目: 北京市自然科学基金-海淀原始创新联合基金(No.L242080); 国家自然科学基金(No.62271086)

中图分类号: TN929.5; TP391.9

文献标识码: A

文章编号: 0372-2112(2025)06-1943-15

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.12263/DZXB.20241053

QoE-Aware Task Offloading in Edge Computing Environment: Potential Game Approach

SHI Zi-qi¹, XIE Gang^{1*}, FAN Wen-hao², LIU Yuan-an²

(1. School of Information and Communication Engineering, Beijing University of Posts and Telecommunication, Beijing 100876, China;

2. School of Electronic Engineering, Beijing University of Posts and Telecommunication, Beijing 100876, China)

Abstract: With the widespread adoption of 5G technology, the importance of edge computing in task offloading and processing has become increasingly prominent. Consequently, this trend has led to the emergence of game theory-based edge computing strategies as a research hot spot. This paper aims to maximize the quality of experience (QoE) by studying the multi-user task offloading problem under time constraints. A system model is constructed from three aspects: communication model, computation model, and time constraints. The optimization problem is first transformed into a game-theoretic problem, and thereafter, the existence of a Nash equilibrium solution is proven. This paper proposes a distributed multi-user offloading (DMUO) algorithm, which enables multiple users to simultaneously update their policies within a single time slot for the first time, significantly reducing computational overhead and improving convergence speed. Theoretical analysis not only demonstrates that the DMUO algorithm converges to the Nash equilibrium solution, but also provides an upper bound on the number of iterations. Furthermore, the robustness of the algorithm is verified by analyzing the performance gap between the worst-case strategy and the optimal solution. Simulation results show that the DMUO algorithm exhibits excellent convergence and system performance, proving its scalability and practical applicability in large-scale edge computing environments.

Key words: mobile edge computing; potential game; Nash equilibrium; computation offloading; quality of experience

Foundation Item(s): Beijing Natural Science Foundation Haidian Original Innovation Joint Fund (No.L242080); National Natural Science Foundation of China (No.62271086)

1 引言

随着信息技术的发展,移动设备上的应用程序会产生更多的任务需要被处理.由于移动设备的限制,仅靠移动设备自身难以处理这些任务.由于公共云距离移动用户较远,如果将任务传输到资源丰富的远程公共云来卸载计算^[1],则会导致系统延迟较大,并且会增大核心链路的网络带宽的压力,所以要求所有移动设备都直接与云交互是不现实的^[2].

为解决这一问题,业界提出了移动边缘计算技术^[3].边缘服务器部署在网络末端与核心网络之间,其通信延迟远低于云端.通过将计算、存储和控制功能前移至更贴近用户的位置,边缘计算使得移动用户能够将任务卸载至边缘节点^[4],从而大幅降低延迟、节省带宽,并增强资源受限设备的计算及安全性能.此外,在网络不稳定或中断的情况下,它还能保持服务的连续性,适用于自动驾驶、无人机和工业控制等场景,支持实时数据处理和响应.未来,边缘计算将与云端紧密协同,构建自云端至终端的服务连续体系,通过智能化、本地化的处理推动新型商业模式的发展,提升系统安全性,并为数据处理提供灵活且高效的解决方案^[5].

在边缘计算领域,任务卸载策略通常可分为部分卸载和完全卸载两类.部分卸载策略将用户的计算任务划分为多个子任务,并将部分子任务卸载至边缘服务器执行.例如,文献[6]提出了一种部分计算卸载方法,通过将计算需求分配到本地设备和边缘服务器,显著降低了任务完成的总延迟.另一方面,完全卸载策略则将整个计算任务迁移至边缘服务器处理,适用于计算密集型任务或本地资源受限的场景.文献[7]针对完全卸载问题,提出了一个基于博弈论的计算卸载模型,模拟物联网用户之间对边缘资源的竞争行为,实现了边缘节点有限处理能力的高效分配.此外,文献[8]以最小化时延为目标,对比研究了完全卸载、部分卸载以及支持设备到设备(Device-to-Device, D2D)通信的部分卸载三种方案.这些研究说明任务卸载策略的适用性取决于具体的任务需求、资源可用性以及网络环境.

关于边缘计算的控制算法可以分为集中式方法和分布式方法.为了满足各种应用的需求,文献[9~12]采用了集中式算法来制定用户的卸载策略.然而这些集中式算法由于其计算复杂性较高,难以适应爆炸性的网络规模^[13],由于分布式算法可以根据各节点的局部信息做出决策,因此使用分布式算法可以有效降低计算的规模.文献[13~20]采用了分布式算法来制定用户的卸载策略.文献[14]以最小化用户总能耗为目标,提出了一种基于交替方向乘子法的分布式算法,以较低的计算复杂度逼近最优解.文献[13, 16~22]则使用博弈论方法研究任务卸载问题.

对于设计分布式算法而言,博弈论是一种有效的数学工具.因为不同参与者有不同的目标,使用博弈论可以让每位参与者都能够根据当前状态做出使自身利益最大化的决策^[23].博弈论中存在不同的博弈模型.文献[16]使用Stackelberg博弈模型,将边缘服务器作为领导者,为本地计算定价,将无线网用户作为跟随者,有效降低了边缘服务器的能耗成本.文献[17]基于协作博弈模型,通过共享助手的空闲资源来实现协作计算,充分利用了异构设备的通信和计算资源.

势博弈是一类特殊的博弈模型.势博弈通过引入势函数,将所有参与者的收益变化统一反映到一个全局目标函数上.这一特性不仅简化了均衡分析,还使得势博弈在分布式系统中表现出良好的稳定性和收敛性.文献[18]首先证明最小化系统计算开销的集中式优化问题是NP(Nondeterministic Polynomial time)困难的,然后提出基于势博弈的分布式卸载算法,展现了卓越的计算卸载性能.文献[19]结合边缘云和公共云服务器,提出一种基于势博弈和贪婪算法的卸载方案,首先通过势博弈确定设备和边缘服务器的初步策略,再用贪婪算法比较公共云方案,从而选择成本最低的策略.文献[20]也研究端-边-云架构的卸载问题,设定通信干扰和资源竞争上界约束:超出通信上界时用本地计算,超出资源上界时卸载到公共云,否则选择边缘云,再用势博弈选择卸载信道.

文献[18~20]均基于势博弈的有限改进特性,通过假设其他用户的策略固定不变,逐步优化单个用户的策略,直至所有用户的策略达到不可进一步改进的状态,从而收敛至纳什均衡解.然而,这种单用户串行更新的方式在单时隙内仅能优化一个用户的策略,导致算法的时间效率较低.针对这一问题,本文首次提出了一种创新性算法,允许多个用户在单时隙内同时更新策略,从而显著提升了算法的执行效率.这一改进不仅提高了算法的实用性,还为大规模边缘计算环境中的任务卸载问题提供了更高效的解决方案.

势博弈在边缘计算中起着至关重要的作用,随着边缘计算的广泛应用,用户体验质量(Quality of Experience, QoE)作为衡量用户满意度的重要指标,备受关注^[24].在一些研究中,QoE被作为优化目标,旨在提升用户在使用边缘计算服务时的体验.例如,文献[25]提出了一种新的QoE模型,同时考虑服务时延、能耗以及任务成功率,并提出一种基于深度强化学习的算法提高了QoE的性能.文献[26]从应用程序供应商的角度解决用户分配问题,根据服务质量将QoE分为不同的级别,但是忽略了边缘计算对于实时性的要求.这些研究揭示了在边缘计算中综合考虑QoE的必要性,为进一步提升边缘计算的用户体验质量提供了理论基础和

方法支持.

基于上述研究,本文进一步探讨了完全卸载策略在优化用户体验质量和资源利用方面的潜力,以最大化用户体验质量为目标,重点研究了时间约束条件下的多用户任务卸载问题.本文首先描述了一个考虑时间约束对卸载决策的影响的优化问题,每个用户的 QoE 会受到其他用户策略的影响.然后我们给出并证明了存在纳什均衡解,并提出了一种分布式多用户卸载算法(Distributed Multi-User Offloading, DMUO),该算法的核心创新在于,允许单时隙内多个用户根据当前环境下的更优策略同时更新卸载决策,从而显著提升算法的执行效率.最后,证明该算法能够收敛到纳什均衡解并给出迭代次数的上限,并分析最坏情况下得到的策略性能与最优解的差距.本文的主要贡献如下:

(1)本文以最大化用户体验质量为目标,研究了时间约束条件下的多用户任务卸载问题.本文从通信模型、计算模型和时间约束三个方面建立系统模型.用户的任务可以在用户设备上本地处理,或者通过无线信道卸载到边缘服务器上处理.

(2)每个用户的目标都是最大化自身 QoE,本文把优化问题建模为一个多用户博弈,提出了一种分布式多用户卸载算法(DMUO),通过证明该博弈是一个势博弈,给出并证明存在纳什均衡解.

(3)DMUO 算法创新性地实现了单时隙内多用户策略的同步更新,与现有基于势博弈的算法相比,显著减少了所需的时隙数量.仿真结果表明,随着用户和系统规模的增加,该算法仍然具有良好的收敛性和系统性能,验证了其在大规模边缘计算环境中的有效性和实用性.

2 系统模型

本文假设一个无线基站覆盖范围内有 N 个用户,用户集合由 $\mathcal{N} = \{1, 2, 3, \dots, N\}$ 表示,存在 K 个信道,信道集合由 $\mathcal{K} = \{1, 2, 3, \dots, K\}$ 来表示.假设每个用户都有一个计算密集型任务要完成,将任务表示为 $\chi_n \triangleq (B_n, D_n)$, B_n 表示计算输入数据的大小, D_n 表示完成计算任务所需的 CPU (Central Processing Unit) 周期的总数.由于完全卸载能够将任务分配至最适配的计算环境,从而避免兼容性问题,同时规避部分卸载可能带来的额外能耗缺陷^[27],本文聚焦于完全卸载问题的研究.在该框架下,每个用户可以选择在本地设备上执行计算任务,也可以通过选择合适的无线信道将任务整体卸载至边缘服务器进行计算^[28].例如文献[29, 30],本文考虑了一种准静态场景,其中移动设备用户集在计算卸载期内保持不变,但可能在不同时期发生变化.

将用户 n 的卸载决策定义为 $a_n, a_n \in \{0\} \cup \mathcal{K}$. 如

图 1 所示,用户不仅可以选择在本地进行计算,还可以选择通过无线通信,将任务卸载到边缘服务器进行边缘云计算.如果 $a_n = 0$,代表用户选择本地计算任务;如果 $a_n = k$,说明用户通过第 k 个信道进行边缘云计算.将所有用户的策略集表示为 $a = (a_1, a_2, a_3, \dots, a_N)$.

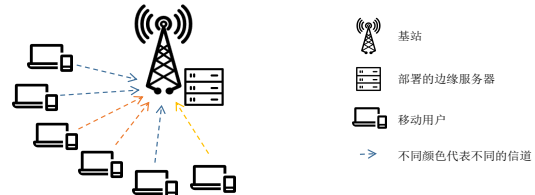


图 1 用户通过信道进行边缘计算

2.1 通信模型

本文考虑无线干扰模型,多个用户可能接入同一无线信道.对于任意用户来说,接入同一信道的其他用户的功率可以看作是噪声.

给定所有移动设备用户的决策概况 a ,根据香农公式,可以计算移动设备用户选择通过无线信道将计算卸载到云的上行链路数据速率为

$$r_n(a) = W \times \log_2 \left(1 + \frac{q_n g_n}{w_0 + \sum_{i \in \mathcal{N} \setminus \{n\}; a_i = a_n} q_i g_i} \right) \quad (1)$$

其中, q_n 和 g_n 分别表示用户 n 的发射功率和信道增益, w_0 表示白噪声, W 表示信道带宽^[31]. $\sum_{i \in \mathcal{N} \setminus \{n\}; a_i = a_n} q_i g_i$ 表示用户在无线通信中受到其他用户的影响.因为需要处理的原始数据量通常远大于结果数据量,本文只考虑上行链路的开销^[32,33].

2.2 计算模型

接下来,将介绍本地计算和边缘云计算方法在能耗和处理时间方面的假定成本.

2.2.1 本地计算模型

对于本地计算方法,移动设备用户在移动设备上本地执行其计算任务,此时 $a_n = 0$.移动设备用户的计算能力(即每秒 CPU 周期)设为 f_n^m .本地计算不存在传输时延,计算执行时间为 $t_n^m = D_n / f_n^m$,计算能量为 $e_n^m = \kappa_n \times D_n$,其中 κ_n 表示用户每个 CPU 周期的能耗因子^[11].

为了综合考虑时间成本和能耗成本^[34],本文将本地计算成本定义为

$$z_n^m = \lambda_n^t t_n^m + \lambda_n^e e_n^m \quad (2)$$

其中, $\lambda_n^t, \lambda_n^e \in [0, 1]$ 分别表示用于移动设备用户决策的时间和能量的加权参数.

2.2.2 边缘云计算模型

对于云计算方法,移动设备用户将通过无线接入将其计算任务卸载到电信运营商部署的边缘节点,此时 $a_n > 0$. 可得传输时间为 $t_{n,\text{off}}^c(a) = B_n/r_n(a)$, 计算时间为 $t_{n,\text{exe}}^c = D_n/f_n^c$, 时间成本为 $t_n^c(a) = t_{n,\text{off}}^c(a) + t_{n,\text{exe}}^c$. 传输能耗 $e_n^c(a) = q_n B_n/r_n(a)$ 是可以根据计算时间和能量计算的开销,边缘云计算成本如下所示:

$$z_n^c(a) = \lambda_n^t t_n^c(a) + \lambda_n^e e_n^c(a) \quad (3)$$

2.3 时间约束模型

为了满足实时计算的要求,用户可以根据自身需求,设置一个时间阈值参数 η_n 来判断用户是否可以通过边缘计算执行计算密集型任务. 当边缘云计算的时间成本大于 $\eta_n t_n^m$ 时,说明不能满足实时计算的要求,应选择本地计算. 即当用户将任务通过无线接入卸载到边缘云时,应满足以下条件:

$$t_n^c(a) < \eta_n t_n^m \quad (4)$$

2.4 用户 QoE 模型

在本文中,考虑每个设备生成一个任务,并关注任务的卸载策略. 由式(2)和式(3)可知,用户的成本可以表示为

$$c_n(a) = \begin{cases} z_n^c(a), & a_n > 0 \\ z_n^m, & a_n = 0 \end{cases} \quad (5)$$

对于任意一个用户,都存在计算成本的上限和下限. 本地计算成本的上限表示为 $z_{\text{max}}^m = \max\{z_n^m\}$, 下限表示为 $z_{\text{min}}^m = \min\{z_n^m\}$. 每个用户的信息传输速率的上限和下限分别表示为式(6)和式(7):

$$r_n^{\text{max}} = W \times \log_2 \left(1 + (q_n g_n)/w_0 \right) \quad (6)$$

$$r_n^{\text{min}} = W \times \log_2 \left(1 + (q_n g_n) / \left(w_0 + \sum_{i \in \mathcal{N} \setminus \{n\}} q_i g_i / K \right) \right) \quad (7)$$

所有用户的信息传输速率的上限和下限可以表示为 $r_{\text{max}} = \max_{n \in \mathcal{N}} \{r_n^{\text{max}}\}$ 和 $r_{\text{min}} = \min_{n \in \mathcal{N}} \{r_n^{\text{min}}\}$. 云计算时间的上限和 下 限 为 $t_{\text{max}}^c = B_{\text{max}}/r_{\text{min}} + D_{\text{max}}/f_{\text{min}}^c$ 和 $t_{\text{min}}^c = B_{\text{min}}/r_{\text{max}} + D_{\text{min}}/f_{\text{max}}^c$. 云计算能耗上限和 下 限 表 示 为 $e_{\text{max}}^c = q_{\text{max}} B_{\text{max}}/r_{\text{min}}$ 和 $e_{\text{min}}^c = q_{\text{min}} B_{\text{min}}/r_{\text{max}}$. 其中 $B_{\text{max}} = \max_{n \in \mathcal{N}} \{B_n\}$, $B_{\text{min}} = \min_{n \in \mathcal{N}} \{B_n\}$ 分别表示最大计算输入数据和最小计算输入数据,而 $D_{\text{max}} = \max_{n \in \mathcal{N}} \{D_n\}$, $D_{\text{min}} = \min_{n \in \mathcal{N}} \{D_n\}$ 分别表示完成计算任务所需的最大 CPU 周期总数和最小周期总数.

对于任意用户来说,通过无线接入的边缘云计算成本满足:

$$z_n^c(a) = \lambda_n^t t_n^c(a) + \lambda_n^e e_n^c(a) \leq (\lambda_n^t + \lambda_n^e) \max\{t_{\text{max}}^c, e_{\text{max}}^c\} = \max\{t_{\text{max}}^c, e_{\text{max}}^c\} \quad (8)$$

因此云计算成本存在上限,同理可知云计算成本存在下限. 由式(5)可知,用户的最大和最小成本为 $c_{\text{max}} = \max\{z_{\text{max}}^m, z_{\text{max}}^c\}$ 和 $c_{\text{min}} = \min\{z_{\text{min}}^m, z_{\text{min}}^c\}$.

QoE 和用户成本表现出非线性负相关性^[26]. 如图2所示,当成本增加时,QoE 将降低. 当成本降低到特定水平时,无论用户成本是否显著减少,用户的 QoE 都显示出非常微不足道的改善;当用户成本在本地计算成本附近变化时,用户的 QoE 变化较大. 根据式(4)和式(5),在本文中,用户的 QoE 定义如下所示:

$$Q_n(a) = \begin{cases} \rho(1+\beta), & a_n = 0 \\ \rho \left(\frac{2}{1+f(a)} + \beta \right) \cdot I(t_n^c(a) < \eta_n t_n^m), & a_n > 0 \end{cases} \quad (9)$$

其中, $\rho = 1 / \left(\frac{1}{1+e^{-\mu}} - \frac{1}{1+e^{\mu}} \right) > 0$, $\beta = -2 / (1+e^{\mu})$, $f(a) =$

$\exp \left(\mu \frac{z_n^c(a) - z_n^m}{c_{\text{max}} - c_{\text{min}}} \right)$. μ 是一个大于 0 的常系数,可以控制 QoE 的增长率. $I(\cdot)$ 是一个示性函数,当 $t_n^c(a) < \eta_n t_n^m$ 时, $I(t_n^c(a) < \eta_n t_n^m) = 1$, 否则 $I(t_n^c(a) < \eta_n t_n^m) = 0$.

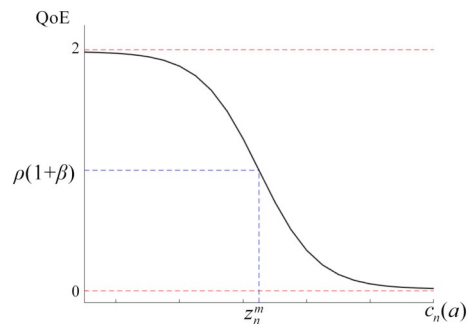


图2 QoE 和用户成本关系

2.5 问题描述

本文将除用户之外的所有其他用户的计算卸载决策定义为 a_{-n} . 本文的优化目标是在当前环境已知且具有时间约束的环境下,每个用户可以通过选择合适的策略,最大化自身的 QoE,即

$$\begin{aligned} \max_{a_n \in \{0\} \cup \mathcal{K}} Q_n(a_n, a_{-n}), \quad \forall n \in \mathcal{N} \\ \text{s.t.} \quad \text{式(4)} \end{aligned} \quad (10)$$

用户之间存在竞争关系,所有用户都争夺有限的资源,希望获得更多的资源来最大限度地提高自己的 QoE. 因此,所有相互竞争的用户很难达到稳定和平衡的状态.

本文将式(10)看作为一个策略博弈 $\zeta =$

$(\mathcal{N}, \{\mathcal{A}_n\}_{n \in \mathcal{N}}, \{Q_n(a)\}_{n \in \mathcal{N}})$, \mathcal{N} 是所有用户的集合, $\mathcal{A}_n \triangleq \{0\} \cup \mathcal{K}$ 表示用户可以选择的策略集合, $\{Q_n(a)\}_{n \in \mathcal{N}}$ 是所有用户的 QoE 函数的集合. 在该博弈中, 每个用户的目标是最大化自身的 QoE 函数.

3 算法设计与分析

由于现有的基于势博弈的分布式算法在每个时隙中只能更新一个用户的策略, 所以使用现有算法需要较多的等待时隙数量, 会造成时间的浪费. 本文对现有的算法进行改进, 提出了新的分布式算法 DMUO, 该算法首次实现了在一个时隙中多个用户同时更新策略, 极大地减少了算法所需的时隙数量, 并且具有良好的系统性能.

3.1 分布式多用户卸载算法 (DMUO)

DMUO 算法如算法 1 所示.

算法 1 DMUO 算法

输入: 用户的任务 χ_n 、传输功率 p_n 和信道增益 g_n

输出: 所有用户的策略集合 $a(t)$

1. 初始化 $t=0, a_n(t)=0$
2. WHILE 没有接收到 STU 信号
3. FOR 用户 n IN 集合 \mathcal{N}
4. 计算用户 n 采取策略 $a_n(t)$ 时的 QoE, 即 $Q_n(a_n(t), a_{-n}(t))$
5. 寻找用户 n 的更优策略 $a_n^b(t)$, 使 $Q_n(a_n^b(t), a_{-n}(t)) > Q_n(a_n(t), a_{-n}(t))$
6. IF 用户 n 存在更优策略 $a_n^b(t)$
7. 将用户 n 放入矩阵 $G(t)$ 的第 $a_n(t)$ 行第 $a_n^b(t)$ 列
8. END IF
9. END FOR
10. IF 矩阵 $G(t)$ 为空矩阵
11. 发送 STU 信号
12. ELSE
13. WHILE 矩阵 $G(t)$ 不为空矩阵
14. 从 $G(t)$ 中任选一个用户 i 放入集合 $\mathcal{U}(t)$, 更新用户 i 的策略 $a_i(t+1) = a_i^b(t)$
15. 将 $G(t)$ 第 $a_i(t)$ 行、第 $a_i^b(t)$ 行、第 $a_i(t)$ 列、第 $a_i^b(t)$ 列的元素置为空
16. END WHILE
17. 未更新策略的用户保持原有策略 $a_n(t+1) = a_n(t)$
18. END IF
19. 时隙 $t=t+1$
20. END WHILE
21. RETURN 所有用户的策略 $a(t)$

首先进行初始化. 将迭代开始时间设置为 0 即 $t=0$, 将所有用户初始化为本地计算. 将在第 t 个时间间隙内, 用户 $n \in \mathcal{N}$ 采取的策略记为 $a_n(t)$. 矩阵 $G(t)$ 是可更新用户矩阵, 初始化为 $K+1$ 行 $K+1$ 列的空矩阵.

然后对所有用户进行遍历, 寻找在第 t 个时间间隙内, 每个用户使自身 QoE 增大的更优策略, 记为 $a_n^b(t)$, 其定义为式 (11). 如果用户 n 找不到更优策略, 则说明当前已经采取了最优的策略, 因此不需要改进策略. 否则, 说明用户 n 没有采取最优的策略, 将用户放入矩阵 $G(t)$ 的第 $a_n(t)$ 行第 $a_n^b(t)$ 列, 如图 3 所示, 如果用户 i 在第 t 个时间间隙内通过第 3 个信道在边缘云上卸载即 $a_i(t)=3$, 而它的更优策略是通过第 2 个信道卸载即 $a_i^b(t)=2$, 则用户应被放入矩阵的第 3 行第 2 列.

$$a_n^b(t) \triangleq \arg_{a_n \in \mathcal{A}_n \setminus \{a_n(t)\}} (Q_n(a_n, a_{-n}(t)) > Q_n(a_n(t), a_{-n}(t))) \quad (11)$$

对所有用户遍历后, 如果矩阵 $G(t)$ 为空矩阵, 说明所有用户都没有更优策略, 此时所有用户都达到最优解, 没有动力改变策略 (到达纳什均衡状态), 因此发送 STU 信号, 退出循环, 算法结束.

如果矩阵 $G(t)$ 不为空, 说明至少有一个用户还存在更优的策略, 此时策略集合需要继续改进, 用户将向基站更新信息, 基站会从中选择用户并向他们发送允许更新的信息, 基站从矩阵 $G(t)$ 中随机选择用户 i 进行策略更新, 将其下一次迭代时的策略更新为更优策略, 即式 (12) 所示. 同时将用户 i 放入集合 $\mathcal{U}(t)$ 中, $\mathcal{U}(t)$ 是在第 t 时间段, 更新策略的用户集合, 定义为式 (13).

$$a_i(t+1) = a_i^b(t) \quad \forall i \in \mathcal{U}(t) \quad (12)$$

$$\mathcal{U}(t) \triangleq \{i \in \mathcal{N}: a_i(t+1) \neq a_i(t)\} \quad (13)$$

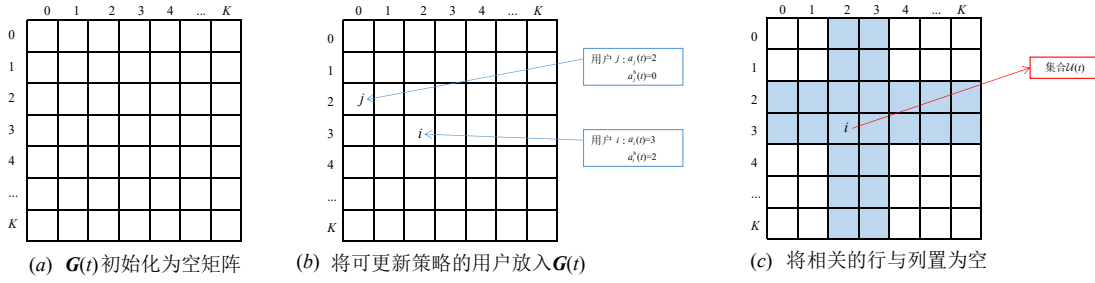
然后将 $G(t)$ 中第 $a_i(t)$ 行、第 $a_i^b(t)$ 行、第 $a_i(t)$ 列、第 $a_i^b(t)$ 列中所有元素置为空. 然后进入下一次循环, 直到变为一个空矩阵. 通过删除与更新策略的用户相关的元素, 可以满足式 (14). 然后记录时间间隙 t , 进行下一次的迭代, 直到接收到 STU 信号, 算法结束.

$$\begin{cases} a_n(t) \neq a_i(t) \\ a_n(t) \neq a_i(t+1) \\ a_n(t+1) \neq a_i(t) \\ a_n(t+1) \neq a_i(t+1) \end{cases}, \forall n \in \mathcal{U}(t), \forall i \in \mathcal{U}(t) \setminus \{n\} \quad (14)$$

3.2 DMUO 收敛性

本节首先证明该博弈存在纳什均衡解, 然后证明 DMUO 可以收敛到纳什均衡点并从理论上推导出最大收敛次数存在上界.

在博弈论中, 若存在一个势函数, 使得当任意一个用户通过单方面改变策略以提升其自身效用函数时, 势函数的值也随之增加, 则该博弈被称为势博弈. 势博弈的一个重要性质是至少存在一个纯策略纳什均衡解^[35]. 势函数的核心思想在于通过量化用户策略选择对全局目标的影响, 将复杂的多用户优化问题转化为

图3 矩阵 $G(t)$ 变化

势函数的最大化问题. 在势博弈中, 每个参与主体通过有限次的迭代决策, 能够逐步逼近最优目标函数, 并最终收敛至势函数的极大值解, 从而确保系统的稳定性和收敛性.

纳什均衡则描述了用户策略的稳定状态, 即在给定其他用户策略的情况下, 任何用户都无法通过单方面改变策略来进一步提高自身收益. 纳什均衡(Nash Equilibrium)描述了博弈中用户策略的稳定状态, 这一概念为分析多用户竞争环境下的策略稳定性提供了理论基础, 并在边缘计算领域得到了广泛应用.

本文将证明博弈 ζ 是一个势博弈, 且势函数为

$$\begin{aligned} \varphi(a) = & -\frac{1}{2} \sum_{n \in \mathcal{N}} q_n g_n \left(\sum_{i \in \mathcal{N} \setminus \{n\}} q_i g_i I(a_i = a_n > 0) \right) \\ & - \sum_{n \in \mathcal{N}} q_n g_n M_n I(a_n = 0) \end{aligned} \quad (15)$$

其中, $M_n = \min\{Y_n, T_n\}$, $Y_n = \frac{q_n g_n}{2^{S_n} - 1} - w_0$,

$$T_n = q_n g_n \left/ \left(2^{\frac{B_n}{W(\eta_n t_n^m - D/f_n^c)}} - 1 \right) - w_0 \right.,$$

$$S = (\lambda_n^t + \lambda_n^c q_n) B_n \left/ \left(W \lambda_n^t \left(\frac{D_n}{f_n^m} - \frac{D_n}{f_n^c} \right) + W \lambda_n^c \kappa_n D_n \right) \right..$$

证明 通过式(1)和式(3)可以看出, 当用户选择通过边缘云卸载任务时, 影响用户策略的主要因素是无线信道上接收到的干扰 $\sum_{i \in \mathcal{N} \setminus \{n\}; a_i = a_n} q_i g_i$. 给定用户策略

a , 和文献[19]中证明定理1同理, 如果用户在所选无线信道上接收到的干扰 $\sum_{i \in \mathcal{N} \setminus \{n\}; a_i = a_n} q_i g_i \leq Y_n$, 则用户可以实

现有益的云计算(选择边缘云卸载的QoE大于本地卸载时的QoE). 同理, 当用户选择通过边缘云卸载任务时, 如果用户在所选无线信道上接收到的干扰 $\sum_{i \in \mathcal{N} \setminus \{n\}; a_i = a_n} q_i g_i \leq T_n$, 则用户可以满足时间约束条件 $t_n^c(a) \leq \eta_n t_n^m$.

假设用户 $n \in \mathcal{N}$ 将策略 a_n 更改为策略 a'_n , 并导致QoE的增加, 即 $Q_n(a'_n, a_{-n}) - Q_n(a_n, a_{-n}) > 0$, 将分为3种

情况证明当 $Q_n(a'_n, a_{-n}) - Q_n(a_n, a_{-n}) > 0$ 时, $\varphi(a'_n, a_{-n}) - \varphi(a_n, a_{-n}) > 0$.

情况1 任意用户 n 从本地计算变为边缘云计算, 即 $a_n = 0, a'_n > 0$.

此时满足条件 $\sum_{i \in \mathcal{N} \setminus \{n\}; a_i = a'_n} q_i g_i < Y_n$, $\sum_{i \in \mathcal{N} \setminus \{n\}; a_i = a'_n} q_i g_i < T_n$, 即 $\sum_{i \in \mathcal{N} \setminus \{n\}; a_i = a'_n} q_i g_i < M_n$.

因此可得

$$\begin{aligned} & \varphi(a'_n, a_{-n}) - \varphi(a_n, a_{-n}) \\ &= -\frac{1}{2} q_n g_n \sum_{i \in \mathcal{N} \setminus \{n\}} q_i g_i I(a_i = a'_n) \\ & \quad - \frac{1}{2} \sum_{i \in \mathcal{N} \setminus \{n\}} q_i g_i q_n g_n I(a_i = a'_n) + q_n g_n M_n \\ &= q_n g_n M_n - q_n g_n \sum_{i \in \mathcal{N} \setminus \{n\}} q_i g_i I(a_i = a'_n) \\ &= q_n g_n \left(M_n - \sum_{i \in \mathcal{N} \setminus \{n\}} q_i g_i I(a_i = a'_n) \right) > 0 \end{aligned} \quad (16)$$

情况2 任意用户 n 从边缘云计算变为本地计算, 即 $a_n > 0, a'_n = 0$.

此时满足条件 $\sum_{i \in \mathcal{N} \setminus \{n\}; a_i = a'_n} q_i g_i > Y_n$ 或者 $\sum_{i \in \mathcal{N} \setminus \{n\}; a_i = a'_n} q_i g_i > T_n$.

当 $\sum_{i \in \mathcal{N} \setminus \{n\}; a_i = a'_n} q_i g_i > Y_n$ 时, 满足条件 $\sum_{i \in \mathcal{N} \setminus \{n\}; a_i = a'_n} q_i g_i > M_n$, 可得

$$\begin{aligned} & \varphi(a'_n, a_{-n}) - \varphi(a_n, a_{-n}) \\ &= q_n g_n \sum_{i \in \mathcal{N} \setminus \{n\}} q_i g_i I(a_i = a_n) - q_n g_n M_n \\ &= q_n g_n \left(\sum_{i \in \mathcal{N} \setminus \{n\}} q_i g_i I(a_i = a_n) - M_n \right) > 0 \end{aligned} \quad (17)$$

当 $\sum_{i \in \mathcal{N} \setminus \{n\}; a_i = a'_n} q_i g_i > T_n$ 时, 同理可证 $\varphi(a'_n, a_{-n}) - \varphi(a_n, a_{-n}) > 0$.

情况3 任意用户 n 从通过一个无线信道进行边缘云计算变为通过另外一个信道进行边缘云计算, 即 $a_n = k, a'_n = k'$.

根据式(1)、式(3)和式(5)可知, $\sum_{i \in \mathcal{N} \setminus \{n\}; a_i = a_n} q_i g_i >$

$\sum_{i \in \mathcal{N} \setminus \{n\}; a_i = a'_n} q_i g_i$ 时可得:

$$\begin{aligned} & \varphi(a'_n, a_{-n}) - \varphi(a_n, a_{-n}) \\ &= -\frac{1}{2} q_n g_n \sum_{i \in \mathcal{N} \setminus \{n\}} q_i g_i I(a_i = a'_n) \\ & \quad - \frac{1}{2} \sum_{i \in \mathcal{N} \setminus \{n\}} q_i g_i q_n g_n I(a_i = a'_n) \\ & \quad + \frac{1}{2} q_n g_n \sum_{i \in \mathcal{N} \setminus \{n\}} q_i g_i I(a_i = a_n) \\ & \quad - \frac{1}{2} \sum_{i \in \mathcal{N} \setminus \{n\}} q_i g_i q_n g_n I(a_i = a_n) \\ &= q_n g_n \left(\sum_{i \in \mathcal{N} \setminus \{n\}; a_i = a_n} q_i g_i - \sum_{i \in \mathcal{N} \setminus \{n\}; a_i = a'_n} q_i g_i \right) > 0 \end{aligned} \quad (18)$$

由式(16)~式(18)可得到结论:博弈 ζ 是一个势博弈. 由于博弈是一个势博弈, 因此这个博弈肯定存在一个纳什均衡解^[35].

本文将在第 t 个时间间隙内, 未更新策略的用户集合定义为

$$\mathcal{E}(t) \triangleq \{i \in \mathcal{N}: a_i(t+1) = a_i(t)\} \quad (19)$$

由式(13)和式(19)可知:

$$\mathcal{E}(t) = \mathcal{N} \setminus \mathcal{U}(t) \quad (20)$$

在第 t 个时间间隙内, 对于任意用户来说, 其他用户的策略改变不会影响到用户的 QoE 值, 即

$$\begin{aligned} & Q_n \left(a_n(t), \{a_i(t)\}_{i \in \mathcal{U}(t)}, \{a_j(t)\}_{j \in \mathcal{E}(t)} \right) \\ &= Q_n \left(a_n(t), \{a_i(t+1)\}_{i \in \mathcal{U}(t)}, \{a_j(t)\}_{j \in \mathcal{E}(t)} \right) \end{aligned} \quad (21)$$

$$\begin{aligned} & Q_n \left(a_n(t+1), \{a_i(t)\}_{i \in \mathcal{U}(t)}, \{a_j(t)\}_{j \in \mathcal{E}(t)} \right) \\ &= Q_n \left(a_n(t+1), \{a_i(t+1)\}_{i \in \mathcal{U}(t)}, \{a_j(t)\}_{j \in \mathcal{E}(t)} \right) \end{aligned} \quad (22)$$

证明 首先证明(21), 分为两种情况: ①用户 n 采取本地计算; ②用户 n 采取边缘云计算.

①用户 n 采取本地计算, $a_n(t) = 0$

由式(6)可知, 此时用户 n 的 QoE 为常数, 因此有

$$\begin{aligned} & Q_n \left(a_n(t), \{a_i(t)\}_{i \in \mathcal{U}(t)}, \{a_j(t)\}_{j \in \mathcal{E}(t)} \right) \\ &= \rho(1+\beta) = Q_n \left(a_n(t), \{a_i(t+1)\}_{i \in \mathcal{U}(t)}, \{a_j(t)\}_{j \in \mathcal{E}(t)} \right) \end{aligned}$$

②用户 n 采取边缘云计算, $a_n(t) \in \mathcal{K}$

由式(14)可知:

$$a_n(t) \neq a_i(t), a_n(t) \neq a_i(t+1).$$

因此用户 n 在无线通信中受到的干扰与用户无关, 可得:

$$\begin{aligned} & r_n \left(a_n(t), \{a_i(t)\}_{i \in \mathcal{U}(t)}, \{a_j(t)\}_{j \in \mathcal{E}(t)} \right) \\ &= W \log_2 \left(1 + \frac{q_n g_n}{w_0 + \sum_{i \in \mathcal{N} \setminus \{n\}; a_i = a_n} q_i g_i} \right) \\ &= r_n \left(a_n(t), \{a_i(t+1)\}_{i \in \mathcal{U}(t)}, \{a_j(t)\}_{j \in \mathcal{E}(t)} \right) \end{aligned}$$

由式(3)、式(5)、式(9)可知:

$$Q_n \left(a_n(t), \{a_i(t)\}_{i \in \mathcal{U}(t)}, \{a_j(t)\}_{j \in \mathcal{E}(t)} \right) = Q_n \left(a_n(t), \{a_i(t+1)\}_{i \in \mathcal{U}(t)}, \{a_j(t)\}_{j \in \mathcal{E}(t)} \right).$$

式(21)得证. 同理, 可证式(22).

本文可以得到结论: DMUO 可以收敛到纳什均衡解, 证明如下.

证明 在第 t 个时隙内, 势函数的变化为

$$\begin{aligned} & \varphi(a(t+1)) - \varphi(a(t)) \\ &= \varphi \left(\{a_n(t+1)\}_{n \in \mathcal{U}(t)}, \{a_i(t+1)\}_{i \in \mathcal{E}(t)} \right) - \varphi \left(\{a_n(t)\}_{n \in \mathcal{U}(t)}, \{a_i(t)\}_{i \in \mathcal{E}(t)} \right) \end{aligned}$$

由式(19)知, 未更新策略的用户坚持原来的策略, 即 $\{a_i(t+1)\}_{i \in \mathcal{E}(t)} = \{a_i(t)\}_{i \in \mathcal{E}(t)}$, 所以有

$$\begin{aligned} & \varphi(a(t+1)) - \varphi(a(t)) \\ &= \varphi \left(\{a_n(t+1)\}_{n \in \mathcal{U}(t)}, \{a_i(t)\}_{i \in \mathcal{E}(t)} \right) - \varphi \left(\{a_n(t)\}_{n \in \mathcal{U}(t)}, \{a_i(t)\}_{i \in \mathcal{E}(t)} \right) \end{aligned}$$

由于 $\mathcal{U}(t)$ 不为空, 不妨假设 $\mathcal{U}(t)$ 中有 d 个元素 ($d \geq 1$), 表示为 $\mathcal{U}(t) = \{h_1, h_1, \dots, h_d\}$, 可得:

$$\begin{aligned} & \varphi(a(t+1)) - \varphi(a(t)) \\ &= \varphi \left(\{a_n(t+1)\}_{n \in \mathcal{U}(t)}, \{a_i(t)\}_{i \in \mathcal{E}(t)} \right) \\ & \quad - \varphi \left(a_{h_1}(t), \{a_n(t+1)\}_{n \in \mathcal{U}(t) \setminus \{h_1\}}, \{a_i(t)\}_{i \in \mathcal{E}(t)} \right) \\ & \quad + \varphi \left(a_{h_1}(t), \{a_n(t+1)\}_{n \in \mathcal{U}(t) \setminus \{h_1\}}, \{a_i(t)\}_{i \in \mathcal{E}(t)} \right) \\ & \quad - \varphi \left(a_{h_1}(t), a_{h_2}(t), \{a_n(t+1)\}_{n \in \mathcal{U}(t) \setminus \{h_1, h_2\}}, \{a_i(t)\}_{i \in \mathcal{E}(t)} \right) \\ & \quad + \varphi \left(a_{h_1}(t), a_{h_2}(t), \{a_n(t+1)\}_{n \in \mathcal{U}(t) \setminus \{h_1, h_2\}}, \{a_i(t)\}_{i \in \mathcal{E}(t)} \right) \end{aligned}$$

$$\begin{aligned}
& -\dots - \varphi \left(\left\{ a_n(t) \right\}_{n \in \mathcal{U}(t) \setminus \{h_d\}}, a_{h_d}(t+1), \left\{ a_i(t) \right\}_{i \in \mathcal{E}(t)} \right) \\
& + \varphi \left(\left\{ a_n(t) \right\}_{n \in \mathcal{U}(t) \setminus \{h_d\}}, a_{h_d}(t+1), \left\{ a_i(t) \right\}_{i \in \mathcal{E}(t)} \right) \\
& - \varphi \left(\left\{ a_n(t) \right\}_{n \in \mathcal{U}(t)}, \left\{ a_i(t) \right\}_{i \in \mathcal{E}(t)} \right)
\end{aligned} \tag{23}$$

由式(12)和式(13)可知:

$$\begin{aligned}
& Q_{h_1} \left(\left\{ a_n(t+1) \right\}_{n \in \mathcal{U}(t)}, \left\{ a_i(t) \right\}_{i \in \mathcal{E}(t)} \right) \\
& - Q_{h_1} \left(a_{h_1}(t), \left\{ a_n(t+1) \right\}_{n \in \mathcal{U}(t) \setminus \{h_1\}}, \left\{ a_i(t) \right\}_{i \in \mathcal{E}(t)} \right) \\
& = Q_{h_1} \left(a_{h_1}(t+1), \left\{ a_n(t+1) \right\}_{n \in \mathcal{U}(t) \setminus \{h_1\}}, \left\{ a_i(t) \right\}_{i \in \mathcal{E}(t)} \right) \\
& - Q_{h_1} \left(a_{h_1}(t), \left\{ a_n(t+1) \right\}_{n \in \mathcal{U}(t) \setminus \{h_1\}}, \left\{ a_i(t) \right\}_{i \in \mathcal{E}(t)} \right) \\
& = Q_{h_1} \left(a_{h_1}(t+1), \left\{ a_n(t) \right\}_{n \in \mathcal{U}(t) \setminus \{h_1\}}, \left\{ a_i(t) \right\}_{i \in \mathcal{E}(t)} \right) \\
& - Q_{h_1} \left(a_{h_1}(t), \left\{ a_n(t) \right\}_{n \in \mathcal{U}(t) \setminus \{h_1\}}, \left\{ a_i(t) \right\}_{i \in \mathcal{E}(t)} \right) \\
& = Q_{h_1} \left(a_{h_1}(t+1), a_{-h_1}(t) \right) - Q_{h_1} \left(a_{h_1}(t), a_{-h_1}(t) \right)
\end{aligned}$$

由式(11)知,更新策略后的用户 QoE 增加,即 $Q_n(a_n(t+1), a_{-n}(t)) - Q_n(a_n(t), a_{-n}(t)) > 0, \forall n \in \mathcal{E}(t)$, 因此可得:

$$\begin{aligned}
& Q_{h_1} \left(\left\{ a_n(t+1) \right\}_{n \in \mathcal{U}(t)}, \left\{ a_i(t) \right\}_{i \in \mathcal{E}(t)} \right) \\
& - Q_{h_1} \left(a_{h_1}(t), \left\{ a_n(t+1) \right\}_{n \in \mathcal{U}(t) \setminus \{h_1\}}, \left\{ a_i(t) \right\}_{i \in \mathcal{E}(t)} \right) \\
& = Q_{h_1} \left(a_{h_1}(t+1), a_{-h_1}(t) \right) - Q_{h_1} \left(a_{h_1}(t), a_{-h_1}(t) \right) > 0
\end{aligned} \tag{24}$$

根据势函数定义,由式(21)可知:

$$\begin{aligned}
& \varphi \left(\left\{ a_n(t+1) \right\}_{n \in \mathcal{U}(t)}, \left\{ a_i(t) \right\}_{i \in \mathcal{E}(t)} \right) \\
& - \varphi \left(a_{h_1}(t), \left\{ a_n(t+1) \right\}_{n \in \mathcal{U}(t) \setminus \{h_1\}}, \left\{ a_i(t) \right\}_{i \in \mathcal{E}(t)} \right) \\
& > 0
\end{aligned} \tag{25}$$

同理可得:

$$\begin{aligned}
& \varphi \left(a_{h_1}(t), \left\{ a_n(t+1) \right\}_{n \in \mathcal{U}(t) \setminus \{h_1\}}, \left\{ a_i(t) \right\}_{i \in \mathcal{E}(t)} \right) \\
& - \varphi \left(a_{h_1}(t), a_{h_2}(t), \left\{ a_n(t+1) \right\}_{n \in \mathcal{U}(t) \setminus \{h_1, h_2\}}, \left\{ a_i(t) \right\}_{i \in \mathcal{E}(t)} \right) \\
& > 0 \\
& \varphi \left(a_{h_1}(t), a_{h_2}(t), \left\{ a_n(t+1) \right\}_{n \in \mathcal{U}(t) \setminus \{h_1, h_2\}}, \left\{ a_i(t) \right\}_{i \in \mathcal{E}(t)} \right) \\
& - \varphi \left(a_{h_1}(t), a_{h_2}(t), a_{h_3}(t), \left\{ a_n(t+1) \right\}_{n \in \mathcal{U}(t) \setminus \{h_1, h_2, h_3\}}, \left\{ a_i(t) \right\}_{i \in \mathcal{E}(t)} \right) \\
& > 0 \\
& \vdots \\
& \varphi \left(\left\{ a_n(t) \right\}_{n \in \mathcal{U}(t) \setminus \{h_d\}}, a_{h_d}(t+1), \left\{ a_i(t) \right\}_{i \in \mathcal{E}(t)} \right) \\
& - \varphi \left(\left\{ a_n(t) \right\}_{n \in \mathcal{U}(t)}, \left\{ a_i(t) \right\}_{i \in \mathcal{E}(t)} \right) \\
& > 0
\end{aligned} \tag{26}$$

将式(25)和式(26)代入式(23),可得:

$$\varphi(a(t+1)) - \varphi(a(t)) > 0 \tag{27}$$

说明随着迭代次数增加,势函数单调递增,又因为存在上界,因此可以收敛到最大值,此时到达纳什均衡点.

接下来将证明,当 P_n, M_n 是正整数时,DMUO 的迭代次数 R_t 存在上限,为 $R_t \leq \sum_{n \in \mathcal{N}} P_n M_n / P_{\min}$, 其中

$$P_n \triangleq q_n g_n, P_{\min} \triangleq \min_{n \in \mathcal{N}} \{P_n\}.$$

证明 因为所有用户初始策略为本地计算,由定理 2 可知,初始用户策略的势函数为 $\varphi(a(0)) = -\sum_{n \in \mathcal{N}} q_n g_n M_n$.

由式(27)可知,随着迭代次数增加,势函数 $\varphi(a(t))$ 单调递增,可得 $\varphi(a(t)) \geq \varphi(a(0))$. 又因为 $\varphi(a(t)) < 0$,可以得到:

$$0 > \varphi(a(t)) \geq -\sum_{n \in \mathcal{N}} q_n g_n M_n = -\sum_{n \in \mathcal{N}} P_n M_n \tag{28}$$

由式(23)和式(26)可知:

$$\begin{aligned}
& \varphi(a(t+1)) - \varphi(a(t)) \\
& \geq \varphi \left(\left\{ a_n(t) \right\}_{n \in \mathcal{U}(t) \setminus \{h_d\}}, a_{h_d}(t+1), \left\{ a_i(t) \right\}_{i \in \mathcal{E}(t)} \right) \\
& - \varphi \left(\left\{ a_n(t) \right\}_{n \in \mathcal{U}(t)}, \left\{ a_i(t) \right\}_{i \in \mathcal{E}(t)} \right)
\end{aligned}$$

因为 h_d 可以是集合 $\mathcal{U}(t)$ 中的任意一个用户,因此

可得:

$$\begin{aligned} & \varphi(a(t+1)) - \varphi(a(t)) \\ & \geq \varphi\left(a_n(t+1), \{a_i(t)\}_{i \in \mathcal{N} \setminus \{n\}}\right) \\ & \quad - \varphi\left(a_n(t), \{a_i(t)\}_{i \in \mathcal{N} \setminus \{n\}}\right), \\ & \quad \forall n \in \mathcal{U}(t) \end{aligned} \quad (29)$$

接下来,将分为3种情况进行讨论.

情况 1 用户 $n \in \mathcal{U}(t)$ 在第 t 个时隙内由本地计算变为边缘云计算,即 $a_n(t) = 0, a_n(t+1) > 0$.

根据式(14)可知:

$$\begin{aligned} & \varphi\left(a_n(t+1), \{a_i(t)\}_{i \in \mathcal{N} \setminus \{n\}}\right) - \varphi\left(a_n(t), \{a_i(t)\}_{i \in \mathcal{N} \setminus \{n\}}\right) \\ & = P_n \left(M_n - \sum_{i \in \mathcal{N} \setminus \{n\}} P_i I(a_i(t) = a_n(t+1)) \right) \end{aligned} \quad (30)$$

因为 M_n, P_i 都是正整数,因此有

$$M_n - \sum_{i \in \mathcal{N} \setminus \{n\}} P_i I(a_i(t) = a_n(t+1)) \geq 1.$$

根据式(16)可知:

$$\varphi\left(a_n(t+1), \{a_i(t)\}_{i \in \mathcal{N} \setminus \{n\}}\right) - \varphi\left(a_n(t), \{a_i(t)\}_{i \in \mathcal{N} \setminus \{n\}}\right) \geq P_n.$$

情况 2 任意用户 $n \in \mathcal{U}(t)$ 从边缘云计算变为本地计算,即 $a_n(t) > 0, a_n(t+1) = 0$.

此时满足条件 $\sum_{i \in \mathcal{N} \setminus \{n\}; a_i = a_n^*} q_i g_i > Y_n$ 或者

$\sum_{i \in \mathcal{N} \setminus \{n\}; a_i = a_n^*} q_i g_i > T_n$. 当 $\sum_{i \in \mathcal{N} \setminus \{n\}; a_i = a_n^*} q_i g_i > Y_n$ 时,由式(17)知:

$$\begin{aligned} & \varphi\left(a_n(t+1), \{a_i(t)\}_{i \in \mathcal{N} \setminus \{n\}}\right) - \varphi\left(a_n(t), \{a_i(t)\}_{i \in \mathcal{N} \setminus \{n\}}\right) \\ & = P_n \left(\sum_{i \in \mathcal{N} \setminus \{n\}} P_i I(a_i(t) = a_n(t)) - M_n \right) \end{aligned}$$

和情况 1 同理,因为 M_n, P_i 都是正整数,可得 $\varphi(a_n(t+1), \{a_i(t)\}_{i \in \mathcal{N} \setminus \{n\}}) - \varphi(a_n(t), \{a_i(t)\}_{i \in \mathcal{N} \setminus \{n\}}) \geq P_n$. 当 $\sum_{i \in \mathcal{N} \setminus \{n\}; a_i = a_n^*} q_i g_i >$

T_n 时,同理可以得到 $\varphi(a_n(t+1), \{a_i(t)\}_{i \in \mathcal{N} \setminus \{n\}}) - \varphi(a_n(t), \{a_i(t)\}_{i \in \mathcal{N} \setminus \{n\}}) \geq P_n$.

情况 3 任意用户 $n \in \mathcal{U}(t)$ 从通过一个无线信道进行边缘云计算变为通过另外一个信道进行边缘云计算,即 $a_n(t) > 0, a_n(t+1) > 0$.

和情况 1 同理,由式(18)可知:

$$\begin{aligned} & \varphi\left(a_n(t+1), \{a_i(t)\}_{i \in \mathcal{N} \setminus \{n\}}\right) - \varphi\left(a_n(t), \{a_i(t)\}_{i \in \mathcal{N} \setminus \{n\}}\right) \\ & = P_n \left(\sum_{i \in \mathcal{N} \setminus \{n\}} P_i I(a_i(t) = a_n(t)) \right. \\ & \quad \left. - \sum_{i \in \mathcal{N} \setminus \{n\}} P_i I(a_i(t) = a_n(t+1)) \right) \\ & \geq P_n \end{aligned}$$

综上,结合式(30)可以得到:

$$\begin{aligned} & \varphi(a(t+1)) - \varphi(a(t)) \\ & \geq \varphi\left(a_n(t+1), \{a_i(t)\}_{i \in \mathcal{N} \setminus \{n\}}\right) \\ & \quad - \varphi\left(a_n(t), \{a_i(t)\}_{i \in \mathcal{N} \setminus \{n\}}\right) \\ & \geq P_n \geq P_{\min} \end{aligned}$$

因此,根据式(29),可知最大迭代次数 R_t 存在上限,为 $\sum_{n \in \mathcal{N}} P_n M_n / P_{\min}$.

3.3 DMUO 算法复杂度分析

由算法可知,DMUO 算法可以分为 3 个部分:用户计算当前策略 QoE、寻找更优的策略和策略更新. 用户计算当前策略 QoE 和时间约束时,仅存在一些基本的数学运算,所以其时间复杂度都可以认为是 $O(1)$. 用户寻找更优的策略时,因为一共可以选择 $K+1$ 个策略,因此该部分的时间复杂度是 $O(2K)$. 在每个决策时隙,基站最多可以选择 $K/2$ 个用户同时更新,选择用户的时间复杂度可以看作是 $O(K/2)$. 当基站向用户发送更新信息后,由于用户在同一个时隙中并行更新策略,更新策略的复杂度可以看作是 $O(1)$. 因此每次迭代的时间复杂度近似为 $O(K)$. 又因为迭代次数存在上限即 $R_t \leq \sum_{n \in \mathcal{N}} P_n M_n / P_{\min}$. 综上,DMUO 算法的时间复杂度为

$$O\left(K \sum_{n \in \mathcal{N}} P_n M_n / P_{\min}\right).$$

3.4 DMUO 算法系统性能分析

博弈可能具有多个纳什均衡解,定义 γ 表示博弈 ζ 的所有纳什均衡解的集合. 为了衡量算法的性能,本文引入了无政府价格 PoA (Price of Anarchy), 其可以衡量最差的纳什均衡解和集中优化最优解之间的差距.

定义 1 令 \tilde{a}_n 表示用户 n 到达纳什均衡时所采取的策略, \tilde{a} 表示到达纳什均衡时的策略集合. 令 a_n^* 表示用户 n 到达集中优化最优解时所采取的策略, a^* 表示到达集中优化最优解时的策略集合.

本文用系统用户 QoE 之和来度量 $\text{PoA}^{[36]}$, 将其定

义为 $\text{PoA} = \min_{\tilde{a} \in \mathcal{T}} \left\{ \sum_{n \in \mathcal{N}} Q_n(\tilde{a}) \right\} / \sum_{n \in \mathcal{N}} Q_n(a^*)$. PoA 表示当用户到达最差的纳什均衡状态时用户 QoE 之和与集中优化最优时的比值. 通过 PoA, 可以从理论上得到算法最差时的性能.

本文将证明 PoA 满足以下不等式:

$$\frac{\sum_{n=1}^N \max \left\{ \rho \left(\frac{2}{1 + e^{\mu \left(\frac{L_i^u - z_{\min}^m}{c_{\max} - c_{\min}} \right)}} + \beta \right), \rho(1 + \beta) \right\}}{\sum_{n=1}^N \max \left\{ \rho \left(\frac{2}{1 + e^{\mu \left(\frac{L_i^o - z_{\max}^m}{c_{\max} - c_{\min}} \right)}} + \beta \right), \rho(1 + \beta) \right\}} \leq \text{PoA} \leq 1.$$

其中, $L_i^u \triangleq \frac{\lambda_i^l B_i + \lambda_i^c q_i B_i}{W \times \log_2 \left(1 + \frac{q_i g_i}{w_0 + \min\{T_i, Y_i\}} \right)} + \lambda_i^t \frac{D_i}{f_i^c}$,

$$L_i^o \triangleq \frac{\lambda_i^l B_i + \lambda_i^c q_i B_i}{W \times \log_2 \left(1 + \frac{q_i g_i}{w_0} \right)} + \lambda_i^t \frac{D_i}{f_i^c}.$$

证明 当所有用户的最佳策略都是本地计算时, 根据式(9), 可知 $Q_n(\tilde{a}) = \rho(1 + \beta) = Q_n(a^*)$, 此时可得 $\text{PoA} = 1$.

接下来, 考虑有用户选择边缘云计算的一般情况, 并假设用户 i 使用边缘云计算, 即 $a_i > 0$. 当用户 i 选择通过一个信道进行传输时, 当没有其他用户对该信道竞争时, 用户 i 的信息传输速率最大, 即 $r_i(a) \leq W \times \log_2 \left(1 + \frac{q_i g_i}{w_0} \right)$. 根据式(3)可知:

$$\begin{aligned} z_i^c(a) &= \frac{\lambda_i^l B_i + \lambda_i^c q_i B_i}{r_i(a)} + \lambda_i^t \frac{D_i}{f_i^c} \\ &\geq \frac{\lambda_i^l B_i + \lambda_i^c q_i B_i}{W \times \log_2 \left(1 + \frac{q_i g_i}{w_0} \right)} + \lambda_i^t \frac{D_i}{f_i^c} = L_i^o. \end{aligned}$$

根据式(9)可知, 此时有

$$\begin{aligned} Q_i(a) &= \rho \left(\frac{2}{1 + e^{\mu \left(\frac{z_i^c(a) - z_{\min}^m}{c_{\max} - c_{\min}} \right)}} + \beta \right) \\ &\leq \rho \left(\frac{2}{1 + e^{\mu \left(\frac{z_i^c(a) - z_{\max}^m}{c_{\max} - c_{\min}} \right)}} + \beta \right) \\ &\leq \rho \left(\frac{2}{1 + e^{\mu \left(\frac{L_i^o - z_{\min}^m}{c_{\max} - c_{\min}} \right)}} + \beta \right) \end{aligned} \quad (31)$$

由于其他用户对用户的干扰存在上限 $\sum_{j \in \mathcal{N} \setminus \{i\}; a_j = a_i} q_j g_j \leq \min\{T_i, Y_i\}$. 因此, 用户的信息传输速率满

足 $r_i(a) \leq W \log_2 \left(1 + \frac{q_i g_i}{w_0 + \min\{T_i, Y_i\}} \right)$. 根据式(3), 可知:

$$\begin{aligned} z_i^c(a) &= \frac{\lambda_i^l B_i + \lambda_i^c q_i B_i}{r_i(a)} + \lambda_i^t \frac{D_i}{f_i^c} \\ &\leq \frac{\lambda_i^l B_i + \lambda_i^c q_i B_i}{W \log_2 \left(1 + \frac{q_i g_i}{w_0 + \min\{T_i, Y_i\}} \right)} + \lambda_i^t \frac{D_i}{f_i^c} \\ &= L_i^u \end{aligned}$$

根据式(9)可知, 此时有

$$\begin{aligned} Q_i(a) &= \rho \left(\frac{2}{1 + e^{\mu \left(\frac{z_i^c(a) - z_{\min}^m}{c_{\max} - c_{\min}} \right)}} + \beta \right) \\ &\geq \rho \left(\frac{2}{1 + e^{\mu \left(\frac{z_i^c(a) - z_{\min}^m}{c_{\max} - c_{\min}} \right)}} + \beta \right) \\ &\geq \rho \left(\frac{2}{1 + e^{\mu \left(\frac{L_i^u - z_{\min}^m}{c_{\max} - c_{\min}} \right)}} + \beta \right) \end{aligned} \quad (32)$$

假设用户 i 选择本地计算时, $Q_i(a) = \rho(1 + \beta)$.

综上, 对于任意用户 n , 满足 $Q_n(a) \leq$

$$\max \left\{ \rho \left(\frac{2}{1 + e^{\mu \left(\frac{L_i^o - z_{\max}^m}{c_{\max} - c_{\min}} \right)}} + \beta \right), \rho(1 + \beta) \right\},$$

因此可得:

$$\sum_{n \in \mathcal{N}} Q_n(a^*) \leq \sum_{n=1}^N \max \left\{ \rho \left(\frac{2}{1 + e^{\mu \left(\frac{L_i^o - z_{\max}^m}{c_{\max} - c_{\min}} \right)}} + \beta \right), \rho(1 + \beta) \right\} \quad (33)$$

对于任意用户 n , 都满足条件 $Q_n(a) \geq$

$$\max \left\{ \rho \left(\frac{2}{1 + e^{\mu \left(\frac{L_i^u - z_{\min}^m}{c_{\max} - c_{\min}} \right)}} + \beta \right), \rho(1 + \beta) \right\},$$

因此可得:

$$\sum_{n \in \mathcal{N}} Q_n(\tilde{a}) \geq \sum_{n=1}^N \max \left\{ \rho \left(\frac{2}{1 + e^{\mu \left(\frac{L_i^u - z_{\min}^m}{c_{\max} - c_{\min}} \right)}} + \beta \right), \rho(1 + \beta) \right\} \quad (34)$$

由式(33)和式(34)可知:

$$\frac{\sum_{n=1}^N \max \left\{ \rho \left(\frac{2}{1 + e^{\mu \left(\frac{L_i^u - z_{\min}^m}{c_{\max} - c_{\min}} \right)}} + \beta \right), \rho(1 + \beta) \right\}}{\sum_{n=1}^N \max \left\{ \rho \left(\frac{2}{1 + e^{\mu \left(\frac{L_i^o - z_{\max}^m}{c_{\max} - c_{\min}} \right)}} + \beta \right), \rho(1 + \beta) \right\}} \leq \text{PoA} \leq 1.$$

可以看出存在上限和下限,也就是说纳什均衡状态下的系统 QoE 和全局最优状态下的 QoE 之间的差距是有限的.

4 仿真分析

本节使用数值分析,对 DMUO 的各项性能进行分析,并与其他方案进行比较. 本文将路径损耗表示为 $15.3 + 37.6 \log_{10}(l_n) \text{dB}^{[37]}$, l_n 是用户设备与基站的距离,基站的覆盖半径设置为 500 m. 其余参数设置见表 1.

表 1 参数设置

参数	数值
用户数量 N	[20,200]
信道数量 K	[4,20]
信道带宽 W/MHz	5
白噪声 w_0/dbm	-100
用户发射功率 q_n/W	100 m
本地 CPU 计算能力 f_n^m/GHz	{0.8,0.9,1}
基站 CPU 计算能力 f_n^s/GHz	10
时间权重 r'_n	{0.3,0.5,0.7}
时间约束 η_n	{0.6,0.8,1}
卸载 CPU 周期 $D_n/\text{Megacycles}$	{900,950,1 000}
卸载数据大小方正汇总行 B_n/kB	{900,950,1 000}

为了评估 DMUO 算法的性能,将其与以下 5 种算法进行比较.

(1)BR. 这是现有的基于势博弈的分布式卸载算法,利用了势博弈的有限改进特性,只有获得更新机会的用户才能根据最佳决策更新策略,单个时隙只能更新一个用户的策略^[18].

(2)SCG. 这种算法首先根据用户的信道增益等性质对用户的优先级进行排序,然后根据排序,每个用户更新自身的策略^[38].

(3)Greedy. 这是一种贪婪算法,每个用户根据当前情况,最大化自身的 QoE^[39].

(4)Random. 每个用户先随机地选择一个信道进行卸载,当 QoE 小于本地计算时,用户使用本地计算.

(5)Local. 所有用户都选择本地计算.

图 4 展示了信道数量对 BR 算法和 DMUO 算法等待时隙的影响. 总体来看,DMUO 算法所需的等待时隙明显少于 BR 算法. 这是因为 BR 算法在每个时隙中仅允许一个用户更新策略,而 DMUO 算法能够在单个时隙内同时更新多个用户的策略,从而显著减少达到纳什均衡解所需的时隙数量. 此外,随着信道数量的增加,算法所需的等待时隙数呈现先增加后减少的趋势. 这种变化可以解释如下:信道数量较少时,增加信道数量为用户提供了更多的策略选择,增加了每次迭代中用户更新策略的可能性,从而提升了算法的复杂度,导致

等待时隙数上升;信道数量较多时,通信资源更加充裕,用户之间的竞争减小,因此等待时隙数随之减少. DMUO 算法由于支持并行策略更新,在效率上相较 BR 算法具有显著优势,而信道数量对等待时隙的影响则体现了资源竞争和策略复杂度之间的动态平衡.

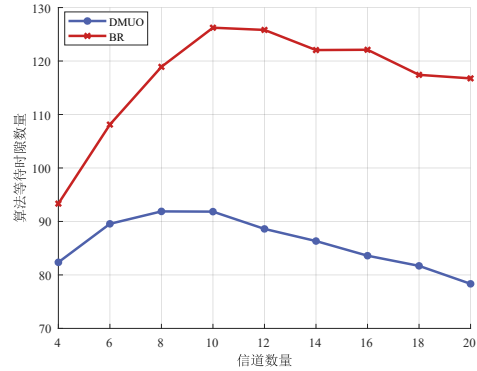


图 4 算法等待时隙随信道数量变化

图 5 表示用户数量分别对 BR 算法和 DMUO 算法的等待时隙的影响. 随着用户数量的增加,DMUO 算法所需的等待时隙数逐渐增加,但增长速度逐渐趋缓. 这是因为用户数量越多,用户之间的竞争越激烈,博弈的复杂度随之提升,从而导致等待时隙的增加. 可以看出 DMUO 算法所需的等待时隙少于 BR 算法,例如,当用户数量为 180 人时,DMUO 算法所需的时隙数量相比 BR 算法减少了 66.1%. 这表明 DMUO 算法在收敛速度上显著优于 BR 算法,并且随着信道数量和用户数量的增加,这一优势变得更加突出.

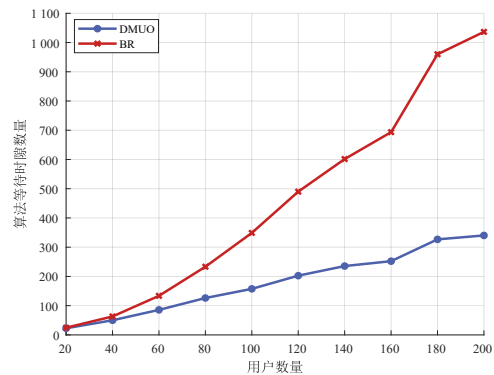


图 5 算法等待时隙随用户数量变化

DMUO 算法与 BR 算法(现有的势博弈算法)均属于基于势博弈的分布式算法. 这两种算法通过与环境的交互获取信息,并在每个时隙中根据获取的信息动态调整用户的策略,最终收敛至纳什均衡解. 然而,DMUO 算法与 BR 算法的核心区别在于用户策略更新的并行性:BR 算法在每个时隙中仅能从矩阵 G 中选择一个用户进行策略更新,而 DMUO 算法则能够从矩阵 G

中选择多个用户同时更新策略. 从算法复杂度分析的角度来看, 用户更新数量并非影响每次迭代时间复杂度的主要因素, 此外每个时隙内用户与基站间的信息传输时延是每个时隙的主要组成部分, 因此, 在时隙级别上, DMUO 算法与 BR 算法在时间开销上表现出相近的特性. 然而, 通过图 4 和图 5 的实验结果可以看出, DMUO 算法的迭代次数显著少于现有的势博弈算法, 这表明 DMUO 算法在整体时间复杂度上优于现有算法. 这一优势主要得益于 DMUO 算法在多用户并行更新策略方面的创新设计, 从而大幅减少了收敛所需的迭代次数. 综上所述, DMUO 算法不仅保留了分布式算法的特性, 还通过引入多用户并行更新机制, 显著提升了算法的执行效率, 为大规模边缘计算环境中的任务卸载问题提供了一种更为高效的解决方案.

图 6 展示了在算法中 10 个用户的 QoE 随时隙的变化情况. 随着时隙的增加, 每个用户的 QoE 会达到一个稳定值, 每个用户都到达当前状态的最大值, 即纳什均衡状态, 从仿真上验证了算法可以收敛到纳什均衡解. 在非平稳过程中一些用户在某个时刻的 QoE 会下降, 这是因为其他用户会对有限的资源进行竞争, 导致当前用户采取策略的成本提高.

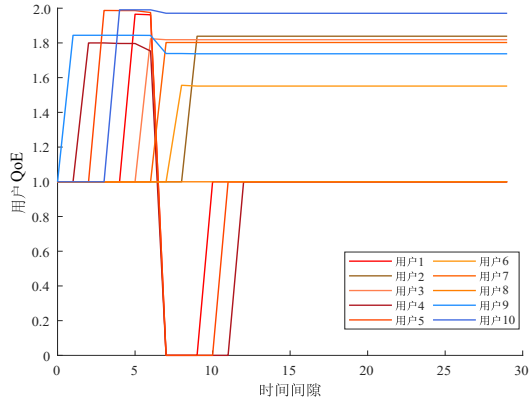


图 6 用户 QoE 随时隙变化

图 7 展示了不同算法下系统 QoE 随时隙变化的情况. 在达到平衡状态后, DMUO 算法的系统 QoE 高于其他算法. 当 DMUO 算法达到平衡状态时, 算法收敛到纳什均衡点, 进一步验证了算法的收敛性.

此外, 由于 DMUO 算法和 BR 算法均能收敛至纳什均衡解, 其最终的系统 QoE 相差不大. 然而, 由于 DMUO 算法允许多个用户在单个时隙内同时更新策略, 其收敛速度显著快于 BR 算法. 在未达到平衡点之前, 系统 QoE 可能会出现短暂波动. 这是因为每个用户在策略选择时仅考虑自身利益最大化, 由于与其他用户的竞争关系, 可能导致系统 QoE 的临时下降.

图 8 展示了不同算法下信道数量变化对系统 QoE

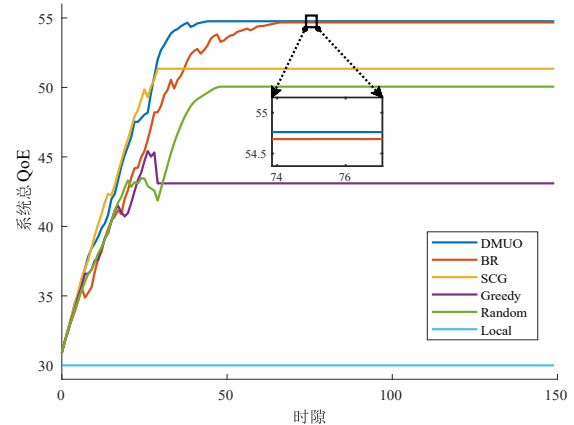


图 7 不同算法的系统总 QoE 随时隙变化

的影响. 从图 8 中可以看出, 除 Local 算法外, 其他算法的系统 QoE 均随着信道数量的增加而显著提升. 这一现象的原因在于, 更多的信道资源为用户提供了更丰富的选择空间, 使其能够根据当前环境状态动态调整卸载策略, 从而优化任务分配并提升系统整体 QoE. 值得注意的是, DMUO 算法在所有信道数量条件下均表现出最优的系统 QoE, 进一步验证了其在资源分配和策略优化方面的显著优势.

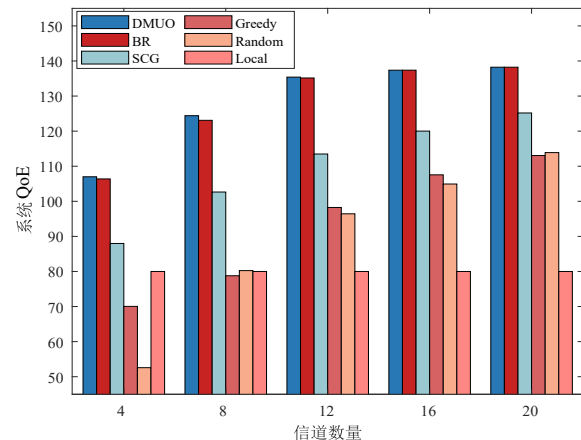


图 8 信道数量对系统总 QoE 的影响

图 9 展示了不同算法下用户数量对系统 QoE 的影响. 随着用户数量的增加, 6 种算法的系统 QoE 均呈现增长趋势, 但其增长幅度存在差异. 具体而言, 用户数量的增加导致资源竞争强度显著提升, 用户间的博弈关系也变得更加复杂. SCG 算法、Greedy 算法、Random 算法和 Local 算法由于未充分考虑用户间的博弈交互, 无法在资源有限的情况下实现高效的资源分配, 其系统 QoE 增长幅度显著低于 DMUO 算法. 随着资源竞争的加剧, 每个用户可分配的通信资源减少, 导致更多用户倾向于选择本地计算而非任务卸载. 然而, DMUO 算法通过高效的博弈模型和多用户并行更新机制, 能够

在资源有限的情况下实现更优的任务分配策略. 当用户数量达到 80 时, DMUO 算法相较于 BR 算法、SCG 算法、Greedy 算法、Random 算法和 Local 算法的系统 QoE 分别提升了 0.6%、22.1%、56.9%、42.4% 和 53.8%. 尽管 DMUO 算法与 BR 算法的系统 QoE 接近, 但 DMUO 算法凭借其更快的收敛速度, 在实际应用中展现出更明显的性能优势. 这些实验结果充分证明了 DMUO 算法的有效性和竞争力.

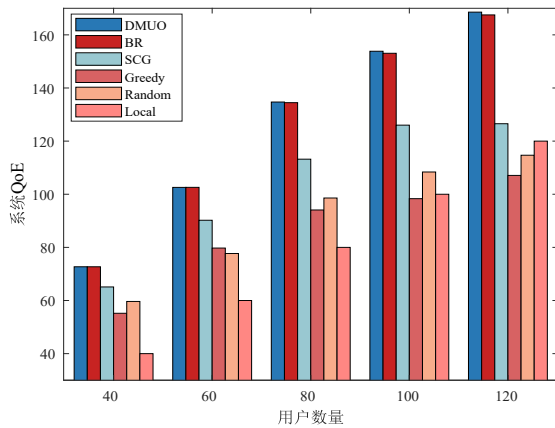


图9 用户数量对系统总 QoE 的影响

图 10 是不同的时间约束条件下, 有益云用户数量随卸载用户数量的变化. 将使用边缘云计算卸载且满足卸载 QoE 大于本地计算 QoE 的用户称为有益云用户, 通过统计有益云用户的数量可以知道系统中用户对公共资源的利用程度. 有益云用户越多, 说明公共资源能使越多的人受益. 如曲线所示, 当过小时, 用户选择策略主要受到时间约束的限制, 此时随着参与博弈的用户数量增加, 有益云用户数量变化较小. 从图 10 中可以看出, 随着用户数量的增大, 不同曲线的距离变小, 这是因为随着用户数量的增大, 对策略选择的约束变小, 此时用户选择策略主要受到通信资源和计算资源的限制.

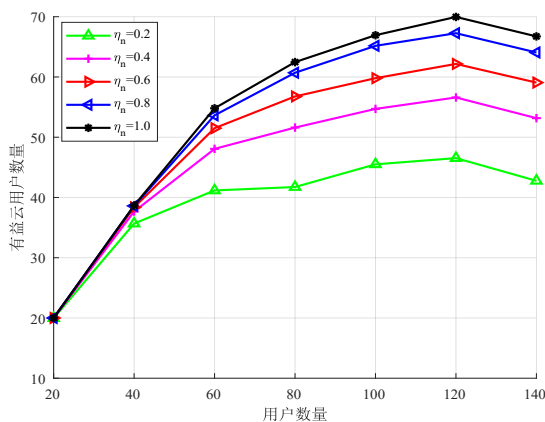


图 10 时间约束对有益云用户数量的影响

5 结束语

本文以最大化用户体验质量为目标, 研究时间约束条件下的多用户任务卸载问题. 本文首先从通信模型、计算模型和时间约束三个方面建立系统模型, 然后给出并证明该博弈中存在纳什均衡解. 本文提出了一种分布式多用户卸载算法 (DMUO), 这种算法首次实现在单时隙内多个用户根据更优策略同时更新策略. 其多用户并行更新机制不仅提升了资源利用效率, 还通过快速收敛降低了资源竞争对系统性能的影响. 最后, 本文证明该算法能够收敛到纳什均衡解并给出迭代次数的上限, 并分析最坏情况下得到的策略性能与最优解的差距. 仿真结果表明, DMUO 算法在大规模用户和高资源竞争强度的场景中仍能保持优异的性能, 进一步证明了其在实际应用中的优越性和实用性.

本文考虑了静态卸载问题, 在未来的工作中, 将考虑动态环境中的用户卸载问题, 即用户状态和信道增益在不同的时隙中可能会发生变化, 通过分析成本的期望, 得到动态环境中的卸载策略. 此外, 本文考虑了信道正交时的任务卸载情况, 将其他用户的干扰看作是噪声, 在未来, 将研究非正交多址 (Non-Orthogonal Multiple Access, NOMA) 技术, 通过使用调制功率和干扰消除 (Successive Interference Cancellation, SIC) 技术, 进一步提高任务卸载的效率.

参考文献

- [1] DINH H T, LEE C, NIYATO D, et al. A survey of mobile cloud computing: Architecture, applications, and approaches[J]. *Wireless Communications and Mobile Computing*, 2013, 13(18): 1587-1611.
- [2] JIN Y, WANG F, ZHAO H W, et al. Survey on trust mechanisms in the environment of cloud computing[J]. *Journal of Chinese Computer Systems*, 2016, 37(1): 1-11.
- [3] MAO Y Y, YOU C S, ZHANG J, et al. A survey on mobile edge computing: The communication perspective[J]. *IEEE Communications Surveys & Tutorials*, 2017, 19(4): 2322-2358.
- [4] DROLIA U, MARTINS R, TAN J Q, et al. The case for mobile edge-clouds[C]//2013 IEEE 10th International Conference on Ubiquitous Intelligence and Computing and 2013 IEEE 10th International Conference on Autonomic and Trusted Computing. Piscataway: IEEE, 2013: 209-215.
- [5] CHIANG M, ZHANG T. Fog and IoT: An overview of research opportunities[J]. *IEEE Internet of Things Journal*, 2016, 3(6): 854-864.
- [6] WU Y, NI K J, ZHANG C, et al. NOMA-assisted multi-access mobile edge computing: A joint optimization of com-

- putation offloading and time allocation[J]. *IEEE Transactions on Vehicular Technology*, 2018, 67(12): 12244-12258.
- [7] SHAH-MANSOURI H, WONG V W S. Hierarchical fog-cloud computing for IoT systems: A computation offloading game[J]. *IEEE Internet of Things Journal*, 2018, 5(4): 3246-3257.
- [8] CAI Y, RAN L, ZHANG J, et al. Latency optimization for D2D-enabled parallel mobile edge computing in cellular networks[J]. *EURASIP Journal on Wireless Communications and Networking*, 2021, 2021(1). DOI: 10.1186/s13638-021-02008-z.
- [9] PU L J, CHEN X, MAO G Q, et al. Chimera: An energy-efficient and deadline-aware hybrid edge computing framework for vehicular crowdsensing applications[J]. *IEEE Internet of Things Journal*, 2019, 6(1): 84-99.
- [10] YANG L, CAO J N, TANG S J, et al. A framework for partitioning and execution of data stream applications in mobile cloud computing[C]//2012 IEEE Fifth International Conference on Cloud Computing. Piscataway: IEEE, 2012: 794-802.
- [11] WEN Y G, ZHANG W W, LUO H Y. Energy-optimal mobile application execution: Taming resource-poor mobile devices with cloud clones[C]//2012 Proceedings IEEE INFOCOM. Piscataway: IEEE, 2012: 2716-2720.
- [12] LI S Y, HU X H, DU Y W. Deep reinforcement learning and game theory for computation offloading in dynamic edge computing markets[J]. *IEEE Access*, 2021, 9: 121456-121466.
- [13] WU L T, SUN P, WANG Z B, et al. Computation offloading in multi-cell networks with collaborative edge-cloud computing: A game theoretic approach[J]. *IEEE Transactions on Mobile Computing*, 2023, 23(3): 2093-2106.
- [14] TANG Q Q, FEI Z S, LI B, et al. Computation offloading in LEO satellite networks with hybrid cloud and edge computing[J]. *IEEE Internet of Things Journal*, 2021, 8(11): 9164-9176.
- [15] NGUYEN D C, PATHIRANA P N, DING M, et al. Deep reinforcement learning for collaborative offloading in heterogeneous edge networks[C]//2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing. Piscataway: IEEE, 2021: 297-303.
- [16] BISHOYI P K, MISRA S. Enabling green mobile-edge computing for 5G-based healthcare applications[J]. *IEEE Transactions on Green Communications and Networking*, 2021, 5(3): 1623-1631.
- [17] LIU Y. Exploiting NOMA for cooperative edge computing[J]. *IEEE Wireless Communications*, 2019, 26(5): 99-103.
- [18] CHEN X, JIAO L, LI W Z, et al. Efficient multi-user computation offloading for mobile-edge cloud computing[J]. *IEEE/ACM Transactions on Networking*, 2016, 24(5): 2795-2808.
- [19] DING Y, LI K L, LIU C B, et al. A potential game theoretic approach to computation offloading strategy optimization in end-edge-cloud computing[J]. *IEEE Transactions on Parallel and Distributed Systems*, 2022, 33(6): 1503-1519.
- [20] CHEN Y, ZHAO J, WU Y, et al. QoE-aware decentralized task offloading and resource allocation for end-edge-cloud systems: A game-theoretical approach[J]. *IEEE Transactions on Mobile Computing*, 2024, 23(1): 769-784.
- [21] HE Q, CUI G M, ZHANG X Y, et al. A game-theoretical approach for user allocation in edge computing environment[J]. *IEEE Transactions on Parallel and Distributed Systems*, 2020, 31(3): 515-529.
- [22] YANG L C, ZHANG H L, LI X, et al. A distributed computation offloading strategy in small-cell networks integrated with mobile edge computing[J]. *IEEE/ACM Transactions on Networking*, 2018, 26(6): 2762-2773.
- [23] ZARISKI O. *An Introduction to Game Theory*[M]. Shanghai: Shanghai University of Finance and Economics Press, 2005.
- [24] FIEDLER M, HOSSFELD T, TRAN-GIA P. A generic quantitative relationship between quality of experience and quality of service[J]. *IEEE Network*, 2010, 24(2): 36-41.
- [25] LU H D, HE X M, DU M, et al. Edge QoE: Computation offloading with deep reinforcement learning for Internet of Things[J]. *IEEE Internet of Things Journal*, 2020, 7(10): 9255-9265.
- [26] LAI P, HE Q, CUI G M, et al. Quality of experience-aware user allocation in edge computing systems: A potential game[C]//2020 IEEE 40th International Conference on Distributed Computing Systems. Piscataway: IEEE, 2020: 223-233.
- [27] 陈宇博. 卫星边缘计算场景下的计算卸载研究[D]. 重庆: 重庆邮电大学, 2022.
- CHEN Y B. Research on Computing Unloading in Satellite Edge Computing Scenarios[D]. Chongqing: Chongqing University of Posts and Telecommunications, 2022. (in Chinese)

- [28] BI S Z, ZHANG Y J. Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading[J]. IEEE Transactions on Wireless Communications, 2018, 17(6): 4177-4190.
- [29] BARBAROSSA S, SARDELLITTI S, DI LORENZO P. Joint allocation of computation and communication resources in multiuser mobile cloud computing[C]//2013 IEEE 14th Workshop on Signal Processing Advances in Wireless Communications. Piscataway: IEEE, 2013: 26-30.
- [30] IOSIFIDIS G, GAO L, HUANG J, et al. An iterative double auction mechanism for mobile data offloading[C]//2013 11th International Symposium and Workshops on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt). Piscataway: IEEE, 2013: 154-161.
- [31] XIAO M B, SHROFF N B, CHONG E K P. A utility-based power-control scheme in wireless cellular systems[J]. IEEE/ACM Transactions on Networking, 2003, 11(2): 210-221.
- [32] KHAN A U R, OTHMAN M, AHMAD MADANI S, et al. A survey of mobile cloud computing application models[J]. IEEE Communications Surveys & Tutorials, 2014, 16(1): 393-413.
- [33] MESSOUS M A, SENOUCI S M, SEDJELMACI H, et al. A game theory based efficient computation offloading in an UAV network[J]. IEEE Transactions on Vehicular Technology, 2019, 68(5): 4964-4974.
- [34] CHEN X. Decentralized computation offloading game for mobile cloud computing[J]. IEEE Transactions on Parallel and Distributed Systems, 2014, 26(4): 974-983.
- [35] MONDERER D, SHAPLEY L S. Potential games[J]. Games and Economic Behavior, 1996, 14(1): 124-143.
- [36] VOORNEVELD M. Equilibria and approximate equilibria in infinite potential games[J]. Economics Letters, 1997, 56(2): 163-169.
- [37] BUDHIRAJA I, KUMAR N, TYAGI S. ISHU: Interference reduction scheme for D2D mobile groups using uplink NOMA[J]. IEEE Transactions on Mobile Computing, 2022, 21(9): 3208-3224.
- [38] ZENG H Y, ZHU X, JIANG Y F, et al. A green coordinated multi-cell NOMA system with fuzzy logic based multi-criterion user mode selection and resource allocation[J]. IEEE Journal of Selected Topics in Signal Processing, 2019, 13(3): 480-495.
- [39] LAI P, HE Q, CUI G M, et al. Edge User Allocation with Dynamic Quality of Service[M]//Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019: 86-101.

作者简介



师子琦 男,现为北京邮电大学信息与通信工程学院硕士研究生. 主要研究方向为边缘计算以及博弈论.

E-mail: shiziqi@bupt.edu.cn



范文浩 男,博士,现为北京邮电大学电子工程学院副教授. 主要研究方向为边缘计算、通感算智融合网络以及物联网技术.

E-mail: whfan@bupt.edu.cn



谢刚 男,现为北京邮电大学信息与通信工程学院副教授. 主要研究方向为无线通信关键技术.

E-mail: xiegang@bupt.edu.cn



刘元安 男,现为北京邮电大学电子工程学院教授,博士生导师. 主要研究方向为电磁场与微波技术.

E-mail: yuliu@bupt.edu.cn