

# 信任环境下考虑系统公平性的边缘计算卸载策略和资源分配

杨守义, 李富康, 任瑞敏

(郑州大学电气与信息工程学院, 河南 郑州 450001)

**摘要:** 在不可靠信任环境下考虑边缘计算中的公平性问题, 针对多用户多服务器场景研究了边缘计算的卸载决策和资源分配问题。在所提出的实时信任模型基础上, 从用户满意度的角度定义了系统公平性, 建立了一种考虑信任和公平性的联合优化模型, 在此基础上提出了一种基于信任的任务卸载和资源分配(TOA)算法。将原 NP-hard 问题转换为多个确定性优化子问题, 基于回溯算法确定卸载策略, 并采用拉格朗日乘子法、凸优化工具箱和二分迭代搜索法完成对资源配置的优化, 从而实现系统公平性, 提升用户满意度。仿真实验结果表明, 所提算法优于其他基准算法。

**关键词:** 边缘计算; 信任模型; 公平性; 资源分配

中图分类号: TN92

文献标志码: A

DOI: 10.11959/j.issn.1000-436x.2024030

## Edge computing offloading policies and resource allocation considering system fairness in trusted environments

YANG Shouyi, LI Fukang, REN Ruimin

School of Electrical and Information Engineering, Zhengzhou University, Zhengzhou 450001, China

**Abstract:** Considering the fairness problem in edge computing in an unreliable trust environment, the offloading decision and resource allocation problem of edge computing was studied for a multi-user multi-server scenario. Based on the proposed real-time trust model, system fairness was defined from the perspective of user satisfaction, a joint optimization model considering trust and fairness was established, and a trust-based task offloading and resource allocation (TOA) algorithm was proposed on this basis. The original NP-hard problem was converted into multiple deterministic optimization subproblems, the offloading strategy was determined based on the backtracking algorithm, and the optimization of resource allocation was accomplished by using the Lagrange multiplier method, the convex optimization toolbox, and the bifurcated iterative search method, so as to achieve system fairness and enhance user satisfaction. Simulation experimental results show that the proposed algorithm outperforms other benchmark algorithms.

**Keywords:** edge computing, trust model, fairness, resource allocation

### 0 引言

6G 将具有支持不断变化需求的能力。为了更好地应用服务能力, 满足网络边缘的密集计算需求, 对边端设备数据处理效率和服务质量的需求日益提升<sup>[1-3]</sup>。然而, 由于计算存储能力和电池容量

的限制, 移动设备无法在短时间内处理计算量巨大的应用任务。为解决这一问题, 移动边缘计算(MEC, mobile edge computing)得到广泛的研究<sup>[4-5]</sup>, 通过将任务卸载到云服务器或边缘服务器来弥补移动设备的性能不足。与传统的移动云计算相比, 边缘服务器更加贴近用户设备, 能有效减少网络传输时

收稿日期: 2023-09-06; 修回日期: 2023-11-23

基金项目: 国家自然科学基金资助项目(No.U1604159)

Foundation Item: The National Natural Science Foundation of China(No.U1604159)

延, 获得更快的响应速度, 保障业务的实时性需求。

另一方面, 如云游戏、车联网、智能家居等 MEC 任务卸载场景涉及云、边、端大量设备参与的资源供给、数据传输和任务计算。在交互过程中, 一些服务器或用户会选择自私的行为, 这将导致 MEC 的可靠性大大降低<sup>[6]</sup>。所以, 引入权威信任机构对节点进行信任管理, 包括评估和更新终端/服务器的可信度, 降低可信度低的节点被选中的概率以提升 MEC 的可靠性, 在移动边缘计算中扮演重要角色<sup>[7]</sup>。文献[8]提出一种基于智能移动终端信任值的计算卸载任务动态分配博弈方法, 实现信任值可靠任务卸载分配。文献[9]以时间可信、行为可信、资源可信为约束, 构建服务质量可信模型, 解决边缘节点低效率选择问题。文献[10]基于信任评估、过滤和选择提出信任感知的任务卸载框架, 以确保用户选择的资源提供商是可信的。文献[11]针对任务卸载服务请求者和资源提供者可信问题, 提出一种基于综合信任评价的任务卸载策略。上述文献从不同的角度建立信任模型优化卸载问题, 但对用户之间的公平性问题很少考虑。

此外, 如何在不可靠网络中保障系统的公平性也是 MEC 中需要考虑的一个重要问题。当前许多工作<sup>[12-13]</sup>致力于寻求更好的平均体验质量 (QoE, quality of experience), 例如更少的平均时间和能耗。然而, 仅保证平均 QoE 的方案可能会导致对单个用户的严重不公平待遇, 用户的体验质量和边缘服务器的服务质量得不到保障。文献[14]考虑到在多设备的 MEC 卸载的场景下, 将应用程序的任务划分为不同比例的部分, 将优化目标构建为最小-最大化问题, 通过凸优化解决 NP-hard 问题研究公平性卸载。文献[15]提出一种面向公平性计算卸载方案, 结合 Balls and Bins 理论和公平性指数, 实现移动 MEC 中的平衡任务分配。文献[16]建立长期纳什均衡的服务卸载博弈模型, 设计基于李雅普诺夫的比例公平资源调度算法, 实现服务稳定性和公平性之间的权衡。文献[14-16]使用不同方法优化了公平性问题, 但都建立在可靠网络的背景下, 缺少对不可靠网络的研究。

目前, 大多数信任模型仅考虑终端或服务器中的一种信任值, 忽略两者之间的相互信任值, 不同用户对不同服务器的信任值是相互独立的。此外, 相关文献多研究基于信任模型完成任务卸载, 但对于在信任环境下如何保障系统公平的研究较少。本文主要研究信任环境下考虑系统公平性的边缘计

算卸载策略和资源分配, 具体工作如下。

1) 建立实时更新信任模型。由服务器处理任务的占比量定义的直接信任值和由用户对服务器的反馈满意度定义的推荐信任值综合构成服务器信任值。根据用户和服务器的交互行为建立用户-服务器信任。服务器将依据信任值高低提供给用户不同质量的计算资源。

2) 定义用户满意度并用其表示系统的公平性。以优化最差用户满意度为目标, 并满足卸载决策约束、资源池约束、用户发射功率约束、服务器承载约束和服务器计算资源约束等, 构建信任环境下的联合卸载决策和资源分配 (JOSRA, joint offloading strategy and resource allocation) 模型表示系统公平性问题。

3) 该问题是一个多变量耦合的混合整数非线性规划问题, 难以直接求解。本文提出基于信任的任务卸载和资源分配算法, 简称 TOA 算法, 将原问题转化为多个连续子问题。基于动态规划思想将卸载问题转化为多个单背包问题, 使用回溯算法求解。根据博弈论完成资源池的资源分配。最后借助凸优化工具箱和二分迭代搜索算法优化用户发射功率和服务器计算资源分配。

## 1 系统模型

系统模型如图 1 所示, 包含边缘层、终端层和信任机构。边缘层由多个由基站和虚拟服务器组成, 具有无线通信和计算能力, 其计算资源丰富且可实时调整。终端层由多个终端用户组成, 泛指具有通信能力且资源受限的设备, 如手机、车辆、智能家居、平板电脑等。信任机构综合服务器自身行为的直接信任值和用户反馈的推荐信任值更新服务器的实时信任值; 根据用户与服务器之间交互情况管理两者之间的实时信任值。假设终端用户产生不同时延需求的任务需要卸载到边缘服务器上处理, 并假设下载结果数据远小于上传任务数据, 不考虑结果回传时延。

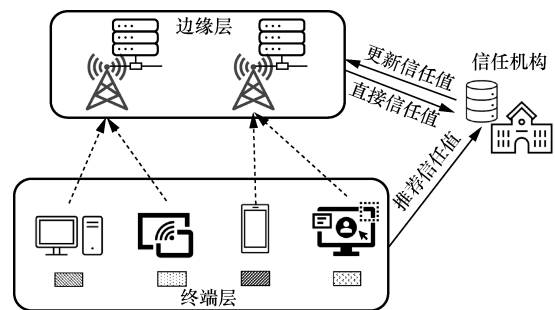


图1 系统模型

### 1.1 终端用户模型

终端用户集合记为  $N = \{1, 2, \dots, N\}, i \in N$ , 用户  $i$  生成任务记为  $\text{Task}_i(t) = \{A_i(t), \lambda_i(t), \tau_i^{\text{tol}}(t)\}$ , 其中  $A_i(t)$  表示任务  $\text{Task}_i(t)$  的数据量大小, 假设其服从 Zipf 分布,  $\lambda_i(t)$  和  $\tau_i^{\text{tol}}(t)$  分别表示任务的时延敏感度和用户  $i$  的最大容忍时延。其中, 最大容忍时延由任务数据量大小、选择服务器的最小计算能力和时延敏感度共同决定, 计算式为

$$\tau_i^{\text{tol}}(t) = \frac{A_i(t)}{\sum_{j=1}^M x_{ij}(t) f_j^{\text{min}}} \lambda_i(t) \quad (1)$$

其中,  $x_{ij}(t)$  是  $t$  时隙的卸载决策,  $f_j^{\text{min}}$  是服务器  $j$  的最小计算资源,  $M$  是服务器总数。

由于终端设备异构, 用户的最大发射功率各不相同, 记为  $\mathcal{P} = \{p_1^{\text{max}}, p_2^{\text{max}}, \dots, p_N^{\text{max}}\}$ , 假设最小发射功率相同, 都为  $p_{\text{min}}$ 。因此可根据系统实时情况调整自身的发射功率以此优化传输时延来保证系统的公平性。假设用户与服务器之间采用正交频分的通信方式, 不考虑用户之间的互相干扰, 则终端用户  $i$  的上行传输速率表示为

$$\text{Rb}_i(t) = \frac{B}{K_{ij}(t)} \text{lb} \left( 1 + \frac{p_i(t) h_{ij}}{N_0} \right) \quad (2)$$

其中,  $B$  是频谱总带宽,  $K_{ij}(t)$  是共享信道的用户数量,  $p_i(t)$  是用户  $i$  的发射功率,  $h_{ij}$  是终端用户到基站的信道增益,  $N_0$  是信道的噪声功率。

所以终端用户  $i$  将任务  $\text{Task}_i(t)$  卸载到基站的传输时延  $\tau_{i,\text{tran}}$  为

$$\tau_{i,\text{tran}}(t) = \frac{A_i(t)}{\text{Rb}_i(t)} \quad (3)$$

### 1.2 虚拟服务器模型

假设基站处配有虚拟服务器构成边缘节点, 负责处理用户任务。虚拟服务器的计算资源量由服务提供商 (SP, service provider) 统一管理调度。服务器集合记为  $\mathcal{M} = \{1, 2, \dots, M\}, j \in \mathcal{M}$ 。其中, 服务器负载上限为  $\text{Bs}_j(t)$ , 当服务器超负载上限时会导致计算能力下降, 记  $\beta_j$  为超载惩罚系数。假设资源池中的计算资源量为  $F$ , 由 SP 分配管理。各服务器异构, 计算资源下限记为  $f_j^{\text{min}}$ , 计算上限  $f_j^{\text{max}}(t)$  则

由 SP 根据每个时隙下服务器的信任值  $\text{rew}_j(t)$  实时更新分配, 服务器  $j$  分配处理任务  $i$  的计算资源记为  $f_{ij}(t)$ , 需要满足约束  $f_j^{\text{min}} \leq \sum_i f_{ij}(t) \leq f_j^{\text{max}}(t)$ 。

任务  $\text{Task}_i(t)$  在服务器上的计算时延  $\tau_{i,\text{com}}(t)$  为

$$\tau_{i,\text{com}}(t) = \frac{A_i(t) S_i}{f_{ij}(t) \text{Tc}_{ij}(t) (1 - \beta_j)} \quad (4)$$

其中,  $\text{Tc}_{ij}(t)$  是信任折扣因子,  $S_i$  是服务器处理 1 bit 任务数据需要的 CPU 周期数。

### 1.3 信任模型

本文参考社会学中的人际关系, 信任是对个体可信行为的一种评估, 而交易频率和交易性质是建立和维护信任关系的 2 个关键要素<sup>[17]</sup>。MEC 环境下的信任和人际关系信任模型有很大的相似性, 本文对边缘服务器的信任值和用户-服务器信任值做出如下定义。

#### 1.3.1 服务器信任值

在每个时隙下, 服务器的信任值受交易频率和交易性质 2 个因素的影响, 将各自影响分别定义为直接信任值和推荐信任值。服务器  $j$  的直接信任值由其处理任务的占比量  $L_j(t)$  所确定, 处理任务的占比量越高, 说明服务器处理数据的容量越大, 对用户任务数据处理的完整性更有保障。  $L_j(t)$  定义式为

$$L_j(t) = \frac{\sum_i x_{ij}(t) A_i(t)}{\sum_i A_i(t)} \quad (5)$$

其中, 分子表示服务器  $j$  处理的任务量, 分母表示所有服务器处理的总任务量。服务器  $j$  的推荐信任值定义为用户对服务器的反馈满意度  $\Psi_j(t)$ , 指接受服务器提供计算服务的用户集合对该服务质量的评分, 评分  $s_i$  由用户  $i$  和服务器  $j$  在时隙  $t$  的信任值  $T_{ij}(t)$  确定。令  $\mathcal{S}_j$  表示服务器  $j$  服务  $k$  个用户得到的服务评分集合, 即  $\mathcal{S}_j = \{s_1, s_2, \dots, s_k\}$ 。根据用户对服务器评分值将集合划分为 2 个子集: 正向评价子集  $\mathcal{S}^P$  和负向评价子集  $\mathcal{S}^N$ 。设定子集分界阈值为  $\xi_T$ , 当用户评分大于阈值时属于正向评价子集, 否则属于负向评价子集, 即

$$\begin{cases} s_i \in \mathcal{S}^N, s_i \leq \xi_T \\ s_i \in \mathcal{S}^P, s_i > \xi_T \end{cases} \quad (6)$$

采用  $\lambda$  和  $\mu$  分别表示正负向评价子集的基数，即  $\lambda = |\mathcal{S}^P|$ ， $\mu = |\mathcal{S}^N|$ ，参考文献[18]，得到服务器的反馈满意度为

$$\psi = \frac{\lambda}{\lambda + \mu} \quad (7)$$

在同一时隙下对服务器的反馈满意度进行归一化处理，使  $\Psi_j \in [0, 1], \forall j \in \mathcal{M}$ ，计算式为

$$\Psi_j(t) = \frac{\psi_j(t)}{\sum_{j=1}^M \psi_j(t)} \quad (8)$$

对直接信任值  $L_j(t)$  和推荐信任值  $\Psi_j(t)$  加权处理得到服务器  $j$  在  $t$  时隙上的瞬时信任值  $r_j(t)$ ，即

$$r_j(t) = \omega_1 L_j(t) + \omega_2 \Psi_j(t), \omega_1 + \omega_2 = 1 \quad (9)$$

考虑服务器的信任值需要通过观测长期行为获得，所以对历史时隙的瞬时信任值进行累积操作，再进行归一化处理得到服务器  $j$  在  $t$  时隙的累积信任值  $\text{rew}_j(t)$ ，即

$$\text{rew}_j(t) = \frac{\sum_{t=1}^T r_j(t)}{\sum_{j=1}^M \sum_{t=1}^T r_j(t)} \quad (10)$$

### 1.3.2 用户-服务器信任值

根据用户和服务器的交互选择行为建立用户-服务器信任值，记为  $T_{ij}(t)$ ，表示在  $t$  时隙的用户  $i$  与服务器  $j$  之间的信任程度。服务器会根据信任值的高低给出不同的服务质量。

考虑用户与服务器之间的行为是快速且不可预测的，所以它们之间的信任是不稳定的，随着时隙推移用户与服务器之间的交互次数的不断增加，信任值会不断发生变化<sup>[19]</sup>。因此必须根据新的交互行为来更新维护信任系统。用户在  $\Delta t$  时隙间隔内的信任值更新表示为

$$T_{ij}^{\text{update}}(t + \Delta t) = T_{ij}(t) + T_{ij}(t + \Delta t) \quad (11)$$

本文通过引入历史信任遗忘函数<sup>[20]</sup>，表示用户

与服务器之间信任值的更新方式。 $\Delta t$  时隙间隔内的信任值更新  $T_{ij}^{\text{update}}(t + \Delta t)$  根据是否选择与上一时隙相同的服务器卸载分为以下 2 种情况。

1) 与上一时隙相同，即  $s^{t-1} = s^t$ ，根据服务器当前信任值和用户-服务器信任值，信任值增加，即  $T_{ij}(t + \Delta t) = \text{rew}_j(t)(1 - T_{ij}(t))$ 。

2) 与上一时隙不相同，即  $s^{t-1} \neq s^t$ ，用户  $i$  与之前选择服务器  $j$  的信任值会有一定程度的下降，即  $T_{ij}(t + \Delta t) = -\text{rew}_j(t)T_{ij}(t)$ 。

结合式(11)，用户  $i$  与服务器  $j$  之间的信任值更新式为

$$\begin{cases} T_{ij}(t+1) = T_{ij}(t) + \text{rew}_j(t)(1 - T_{ij}(t)), s^{t-1} = s^t \\ T_{ij}(t+1) = T_{ij}(t) - \text{rew}_j(t)T_{ij}(t), s^{t-1} \neq s^t \end{cases} \quad (12)$$

服务器根据信任值所处的区间产生不同的信任折扣  $\text{Tc}_{ij}(t)$ 。当服务器承诺分配给用户计算资源量  $f_{ij}(t)$  时，往往因为信任问题而达不到承诺的资源量，即信任值低的服务器对用户可能存在夸大资源量的“欺诈”问题，本文将这种问题模型简化为信任折扣的问题，根据信任值高低划分为多个信任等级，不同等级下的信任折扣不同，计算质量也不同。最终用户得到的计算资源  $f_{ij}^{\text{real}}(t)$  为

$$f_{ij}^{\text{real}}(t) = f_{ij}(t)\text{Tc}_{ij}(t) \quad (13)$$

在建模优化问题时提前考虑到信任折扣问题，以便在后续求解问题的过程中做出相应的补偿处理。

## 2 问题形成

本文旨在信任环境下保证整个系统的长期公平性，考虑完全准确地描述一个动态变化的长期系统是困难的，假设每个时隙内的变量包括用户任务、服务器最大计算资源以及用户-服务器信任值等都是固定不变的。而到下一个时隙信任值更新，新的用户任务随机产生，卸载调度和资源分配都需要重新优化。此外，设有服务器负载上限约束服务器不会承载过多的任务，以防止影响下一时隙的任务处理。因此将各时隙之间简化为相互独立。

通过分时隙处理的方法将长期优化问题转换到每个时隙内求解，即优化每个时隙下系统公平性。首先，定义  $G_i(t)$  表示用户对任务完成情况的满意度，作为衡量系统公平性的一个指标， $G_i(t)$  计算式为

$$G_i(t) = \frac{\tau_i(t)}{\tau_i^{\text{tol}}(t)} \quad (14)$$

其中,  $\tau_i(t)$  是用户  $i$  的实际响应时延,  $\tau_i(t) = \tau_{i,\text{tran}}(t) + \tau_{i,\text{com}}(t)$ 。

每个用户都贪婪地选取信任值高的服务器以获得更多的计算资源, 造成资源分配不均, 进而出现用户满意度参差不齐的不公平现象, 因此本文联合优化卸载策略  $X(t)$ 、用户的发射功率  $p_i(t)$ 、SP 分配给服务器  $j$  的最大计算资源  $f_j^{\text{max}}(t)$  以及服务器  $j$  分配给用户  $i$  的计算资源  $f_{ij}(t)$  多个变量, 将提高最差用户满意度  $\min \max G_i(t)$  作为优化目标, 以保证整个系统的公平性。基于此构建 JOSRA 模型表示系统公平性问题, 表示为

$$\begin{aligned} \text{P1: } & \min_{X(t), p_i(t), f_j^{\text{max}}(t), f_{ij}(t)} \max_{i \in \mathcal{N}} \{G_i(t)\} \\ \text{s.t. C1: } & x_{ij}(t) \in \{0, 1\}, i \in \mathcal{N}, j \in \mathcal{M} \\ \text{C2: } & \sum_{j=1}^M x_{ij}(t) = 1, i \in \mathcal{N} \\ \text{C3: } & \sum_i x_{ij}(t) A_i(t) \leq B s_j(t), j \in \mathcal{M} \\ \text{C4: } & \sum_j f_j^{\text{max}}(t) \leq F, j \in \mathcal{M} \\ \text{C5: } & f_j^{\text{min}} \leq f_j(t) \leq f_j^{\text{max}}(t), j \in \mathcal{M} \\ \text{C6: } & p_i^{\text{min}} \leq p_i(t) \leq p_i^{\text{max}}, i \in \mathcal{N} \\ \text{C7: } & f_{ij}(t) \geq 0, i \in \mathcal{N}, j \in \mathcal{M} \\ \text{C8: } & \sum_i x_{ij}(t) f_{ij}(t) \leq f_j(t), j \in \mathcal{M} \end{aligned} \quad (15)$$

其中, 约束 C1 和 C2 保证用户在一个时隙下只能选择一个服务器进行卸载; C3 是对服务器承载能力的约束; C4 和 C5 是对资源预算的约束, 分别保证分配给虚拟服务器的资源不超过资源池内的总计算资源以及单个服务器资源预算不低于最低资源预算; C6 是用户发射功率的约束; C7 和 C8 保证服务器分配给用户的计算资源不小于零且不超过该服务器的资源预算。

根据目标函数和约束可知, JOSRA 是一个混合整数非线性规划 (MINLP, mixed-integer nonlinear programming) 问题, 属于 NP-hard 问题难以直接求解, 本文将 P1 转换成多个子问题进行求解并提出基于信任的任务卸载和资源分配算法。

### 3 问题求解

本文将 P1 划分为以下 3 个子问题: 服务器最大资源分配、用户卸载决策、通信与计算资源分配。这 3 个子问题是相互关联的, 服务器最大资源决定服务器的承载任务能力, 进而影响用户的卸载决策; 通信与计算资源分配需在前 2 个子问题的基础上求解, 并反过来影响前 2 个子问题。为平衡计算复杂度和计算精度, 本文提出基于信任值的博弈模型求解服务器最大资源预算, 确定该时隙下的服务器承载能力; 然后将卸载问题建模成多背包问题, 根据当前时隙下的信任值和各服务器承载能力来优化卸载决策; 最后, 用迭代方法求解通信与计算资源分配子问题, 将整个时隙的公平性硬解耦为通信和计算 2 个部分来分别优化。

#### 3.1 服务器最大资源分配

SP 首先根据虚拟服务器的信任值配置各自的预算资源。本文考虑服务器自身信任值和计算资源需求制定如下博弈问题

$$\begin{aligned} \text{P2: } & \max_{f_j^{\text{max}}(t)} \prod_{j=1}^M (f_j^{\text{max}}(t) - f_j^{\text{min}})^{\text{rew}_j(t)} \\ \text{s.t. } & \text{C4, C5} \end{aligned} \quad (16)$$

在计算资源的博弈中, 使用服务器的信任值  $\text{rew}_j(t)$  表示议价能力, 对于所有服务器都满足约束  $\sum \text{rew}_j(t) = 1$ 。

假设将资源池内的资源完全分配, 将 C4 转换为等式约束  $\sum_j f_j^{\text{max}}(t) = F$ , 并使用拉格朗日乘子法求解。通过拉格朗日系数  $\sigma$  把等式约束和目标函数组合成如下拉格朗日函数表达式

$$\begin{aligned} L(f_j^{\text{max}}(t), \sigma) = & \sum \text{rew}_j(t) \ln(f_j^{\text{max}}(t) - f_j^{\text{min}}) + \\ & \sigma \left( \sum f_j^{\text{max}}(t) - F \right) \end{aligned} \quad (17)$$

为得到最优解, 对式(17)的单调性和凹凸性进行分析, 其中,  $L$  对  $f_j^{\text{max}}(t)$  的一阶导数和二阶导数分别为

$$\frac{\partial L}{\partial f_j^{\text{max}}(t)} = \frac{\text{rew}_j(t)}{f_j^{\text{max}}(t) - f_j^{\text{min}}} + \sigma \quad (18)$$

$$\frac{\partial^2 L}{\partial f_j^{\max}(t)^2} = -\frac{1}{(f_j^{\max}(t) - f_j^{\min})^2} < 0 \quad (19)$$

由二阶导数小于零可知  $L$  是凹函数, 根据一阶最优条件: 在一阶导数等于零处取得最大值, 令式(18)右侧等于零, 可知 SP 提供给虚拟服务器的最佳资源上限  $f_j^{\max}(t)^*$  为

$$f_j^{\max}(t)^* = f_j^{\min} - \frac{\text{rew}_j(t)}{\sigma} \quad (20)$$

用式(20)中的  $f_j^{\max}(t)^*$  替换式(17)中的  $f_j^{\max}(t)$ , 更新拉格朗日函数表达式如下

$$L = \sum \text{rew}_j(t) \ln\left(-\frac{\text{rew}_j(t)}{\sigma}\right) + \sigma\left(\sum f_j^{\min} - F\right) - 1 \quad (21)$$

同样, 对更新后的拉格朗日函数对另一变量  $\sigma$  求导, 并令  $\frac{\partial L}{\partial \sigma} = 0$  可以得到

$$\sigma^* = \frac{1}{\sum f_j^{\min} - F} \quad (22)$$

将式(22)拉格朗日乘子  $\sigma^*$  代入式(20)替换  $\sigma$ , 可以得到最终服务器计算资源预算的最优解为

$$f_j^{\max}(t)^* = f_j^{\min} + \text{rew}_j(t) \left(-\sum f_j^{\min} + F\right) \quad (23)$$

显然式(23)中有  $\text{rew}_j(t) \geq 0$  且  $\sum f_j^{\min} \leq F$ , 一定有  $f_j^{\max}(t)^* \geq f_j^{\min}$  满足约束 C5, 即所得的最终解  $f_j^{\max}(t)^*$  对于问题 P2 是可行解。SP 根据  $f_j^{\max}(t)^*$  为所有虚拟服务器分配最大计算资源。在给定资源预算  $f_j^{\max}(t)^*$  的情况下, 考虑服务器为不同用户任务提供不同计算资源的支持, 以最大限度地提高资源利用率并保证整个系统的公平性。

### 3.2 用户卸载决策

用户希望将任务卸载到自身信任的服务器上进行处理以得到最大利润, 这里利润定义为用户-服务器信任值, 即  $T_{ij}(t)$ , 该子问题属于广义分配问题 (GAP, generalized assignment problem), 将任务分配给服务器使利润最大化, 同时不超出服务器的承载能力。故将用户卸载决策的 GAP 问题建模成多背包问题, 以用户-服务器信任值  $T_{ij}(t)$  作为物品价值, 以服

务器承载能力  $Bs_j$  作为背包容量约束, 即

$$\begin{aligned} \text{P3: } & \max_{x(t)} x_{ij}(t) T_{ij}(t), i \in N, j \in M \\ \text{s.t. } & \text{C1, C2, C3} \end{aligned} \quad (24)$$

为了降低直接求解 GAP 问题的复杂度, 松弛约束 C2 将 GAP 问题转化成多个连续单背包 0-1 问题求解, 即

$$\begin{aligned} \text{P4: } & Z_j = \max_{x(t)} x_{ij}(t) T_{ij}(t), i \in N \\ \text{s.t. } & \text{C1, C3} \end{aligned} \quad (25)$$

使用动态规划的思想求解单个背包问题, 对服务器  $j$  而言, 令  $z(i, bs)$  表示将前  $i(1 \leq i \leq N)$  个任务卸载到目前剩余容量为  $bs(1 \leq bs \leq Bs_j)$  的服务器  $j$  中获得的利益最大值, 在决策  $x_i$  时已经确定  $(x_1, \dots, x_{i-1})$ , 即前  $i-1$  个任务的卸载方案, 则问题将处于以下 2 种状态。

1) 服务器剩余容量不足以承载任务  $i$ , 则卸载前  $i$  个任务得到的最大价值和卸载前  $i-1$  个任务得到的最大价值是相同的, 即  $x_i = 0$ 。

2) 服务器剩余容量可以承载任务  $i$ , 如果把第  $i$  个任务卸载到服务器, 则服务器处理的业务价值等于把前  $i-1$  个任务卸载到容量为  $bs - A_i(t)$  的服务器获得的价值再加上第  $i$  个任务的价值  $T_{ij}(t)$ ; 如果第  $i$  个任务没有卸载到该服务器, 则服务器处理业务的价值等于把前  $i-1$  个任务卸载到容量为  $bs$  的服务器中所获得的价值。显然, 取两者中价值较大者作为是否将前  $i$  个任务卸载到容量约束为  $bs$  的服务器进行处理的最优解, 可以得到如下最大利益的递推关系式

$$z(i, bs) = \begin{cases} z(i-1, bs), s < A_i(t) \\ \max\{z(i-1, bs), z(i-1, bs - A_i(t)) + T_{ij}(t)\}, s \geq A_i(t) \end{cases} \quad (26)$$

为确定卸载决策, 从  $z(N, Bs_j)$  的值向前推导, 如果  $z(N, Bs_j) > z(N-1, Bs_j)$ , 表示第  $N$  个任务卸载到第  $j$  个服务器, 前  $n-1$  个任务卸载到容量为  $Bs_j - A_N(t)$  的服务器; 否则, 认为第  $N$  个任务没有被卸载到此服务器, 前  $N-1$  个任务被卸载到容量约束为  $Bs_j$  的服务器。依次类推, 直到确定第 1 个任务是否被卸载到此服务器。由此得到服务器  $j$  上

的任务卸载决策函数为

$$x_i(t) = \begin{cases} 0 & , z(i, \text{bs}) = z(i-1, \text{bs}) \\ 1 & , z(i, \text{bs}) > z(i-1, \text{bs}) \end{cases} \quad (27)$$

求解所有服务器上的背包问题, 可得到暂时任务卸载决策  $X'(t)$ , 可能出现以下 2 种情况违背约束 C2。

1) 用户任务卸载到多个服务器处理, 记此类用户集合为  $N^+$ 。用户重复卸载的服务器集合  $M_i^+ = \left\{ j \mid \sum_{j=1}^M x_{ij}(t) \geq 2 \right\}$ , 选择集合中收益最大的服务器进行卸载处理, 即用户  $i \in N^+$  选择卸载的服务器表示为  $j = \left\{ j^* \mid \operatorname{argmax}_{j \in M_i^+} x_{ij}(t) T_{ij}(t) \right\}$ 。

2) 用户任务没有分配到服务器卸载处理, 记此类用户集合为  $N^-$ 。确定集合  $N^+$  中的卸载方案后更新所有服务器的剩余承载容量, 将可以承载用户  $i$  的服务器集合记为  $M_i^- = \left\{ j \mid \sum_{j=1}^M x_{ij}(t) = 0, \text{bs}_j \geq A_i(t) \right\}$ , 优先选择集合  $M_i^-$  中获取利益最大的服务器进行卸载, 即  $j = \left\{ j^* \mid \operatorname{argmax}_{j \in M_i^-} x_{ij}(t) T_{ij}(t) \right\}$ 。

至此, 利用回溯算法求解背包问题获得所有用户在该时隙下基于信任的卸载方案  $X(t)$ 。

### 3.3 通信与计算资源分配

在确定了服务器最大计算资源分配  $f_j^{\max}(t)^*$  和用户卸载决策  $X(t)$  后, 原问题剩余用户发射功率  $p_i(t)$  和服务器分配用户任务的计算资源  $f_{ij}(t)$  待优化, 表示为

$$\begin{aligned} \text{P5: } & \min_{p_i(t), f_{ij}(t)} \max_{i \in N} \{G_i(t)\} \\ \text{s.t. } & \text{C5, C6, C7, C8} \end{aligned} \quad (28)$$

考虑  $p_i(t)$  和  $f_{ij}(t)$  相互耦合, 直接求解困难, 故牺牲部分计算精度进行硬解耦处理, 将两者分别在用户任务传输阶段和服务器处理阶段进行优化, 将问题 P5 拆分成 2 个子问题进行求解。

#### 3.3.1 用户发射功率优化

在优化用户发射功率时, 不考虑计算资源的分配, 只保证任务传输阶段系统的相对公平性。建立子问题如下

$$\begin{aligned} \text{P6: } & \min_{p_i(t)} \max_{i \in N} \frac{\text{lb} \left( 1 + \frac{p_i^{\min} h_{ij}}{N_0} \right)}{\lambda_i(t) \text{lb} \left( 1 + \frac{p_i(t) h_{ij}}{N_0} \right)} \\ \text{s.t. } & \text{C6} \end{aligned} \quad (29)$$

参考文献[21]求解 min-max 问题的方法, 取临时变量  $\text{Tp}_i = \frac{1}{\text{lb} \left( 1 + \frac{p_i(t) h_{ij}}{N_0} \right)}$ , 目标式改写成

$\min_{\text{Tp}_i} \max_{i \in N} \{C_1 \text{Tp}_i\}$ , 令  $G_p^{\max} = \max_{i \in N} \{C_1 \text{Tp}_i\}$  并将目标中最大化问题转换为新约束条件 C9。

$$\text{C9: } C_1 \text{Tp}_i \leq G_p^{\max} \quad (30)$$

又因为  $p_i(t) = \frac{1}{2^{\text{Tp}_i-1} N_0} h_{ij}$ , 且  $\text{Tp}_i$  对  $p_i(t)$  的一阶

导数  $\frac{\partial \text{Tp}_i}{\partial p_i(t)} = -\frac{\left( 1 + \frac{p_i(t) h_{ij}}{N_0} \right) h_{ij}}{\text{lb} \left( 1 + \frac{p_i(t) h_{ij}}{N_0} \right)^2 N_0} < 0$  为单调递减,

所以约束 C6 改写为 C6'。

$$\text{C6': } \frac{1}{\text{lb} \left( 1 + \frac{p_{\max} h_{ij}}{N_0} \right)} \leq \text{Tp}_i \leq \frac{1}{\text{lb} \left( 1 + \frac{p_{\max} h_{ij}}{N_0} \right)} \quad (31)$$

将非凸问题 P6 转换成新拟凸问题 P7。

$$\begin{aligned} \text{P7: } & \min_{\text{Tp}_i} G_p^{\max} \\ \text{s.t. } & \text{C6', C9} \end{aligned} \quad (32)$$

借助凸优化工具箱采用内点法对拟凸问题 P7 求解得到最优的  $\text{Tp}_i^*$ , 通过  $p_i(t) = \frac{1}{2^{\text{Tp}_i^*-1} N_0} h_{ij}$  反解得

到用户以最优发射功率  $p_i^*(t)$ , 保证任务传输部分的公平性。

#### 3.3.2 服务器计算资源分配

为保证整个时隙内系统的公平性, 在传输时延确定的基础上继续优化服务器分配给用户的计算资源  $f_{ij}(t)$ , 建立优化问题 P8 如下

$$\begin{aligned} \text{P8: } \min \max_{f_{ij}(t), i \in N, j \in M} & \frac{\tau_{i,\text{tran}}(t)}{\tau_i^{\text{tol}}(t)} + \frac{C_2}{f_{ij}(t)} \\ \text{s.t. } & \text{C5, C7, C8} \end{aligned} \quad (33)$$

其中,  $C_2 = \frac{A_i(t)S_i}{Tc_{ij}(t)(1-\beta_j)\tau_i^{\text{tol}}(t)}$  是常量部分。令

$$\frac{\tau_{i,\text{tran}}(t)}{\tau_i^{\text{tol}}(t)} + \frac{C_2}{f_{ij}(t)} \leq G_f^{\text{max}}, \text{ 那么原非光滑问题 P8 转换为}$$

$$\begin{aligned} \text{P9: } \min_{f_{ij}(t)} & G_f^{\text{max}} \\ \text{s.t. } & \text{C5, C7, C8} \\ \text{C10: } & \frac{\tau_{i,\text{tran}}(t)}{\tau_i^{\text{tol}}(t)} + \frac{C_2}{f_{ij}(t)} \leq G_f^{\text{max}}, i \in N, j \in M \end{aligned} \quad (34)$$

因为约束 C10 中  $\frac{C_2}{f_{ij}(t)} \geq 0$ , 所以

$$G_f^{\text{max}} - \frac{\tau_{i,\text{tran}}(t)}{\tau_i^{\text{tol}}(t)} \geq 0, \text{ 联合约束 C7 和 C10 可以得到}$$

$$0 \leq \frac{C_2}{G_f^{\text{max}} - \frac{\tau_{i,\text{tran}}(t)}{\tau_i^{\text{tol}}(t)}} \leq f_{ij}(t). \text{ 基于上述分析, 对服务}$$

器上的用户计算资源求和, 并联立约束 C8 和 C5 可以得到不等式

$$\sum_i^N x_{ij}(t) \frac{C_2}{G_f^{\text{max}} - \frac{\tau_{i,\text{tran}}(t)}{\tau_i^{\text{tol}}(t)}} \leq \sum_i^N f_{ij}(t) \leq f_j^{\text{max}}(t) \quad (35)$$

接下来, 需要分配不同服务器上的计算资源给不同的用户, 以提高满意度最差的用户体验来保证整个系统的公平性。为此需要将更多的计算资源分配给满意度最差的用户。因此, 该用户的  $G$  值会降低, 满意度提高; 而其他用户的  $G$  值会升高, 满意度下降。通过循环迭代上述过程, 最终各个服务器将所有计算资源都用于用户任务处理以确保所有用户能获得相似满意度。故在最优条件下可得

$$\sum_i^N x_{ij}(t) \frac{C_2}{G_f^{\text{max}} - \frac{\tau_{i,\text{tran}}(t)}{\tau_i^{\text{tol}}(t)}} = \sum_i^N f_{ij}(t) = f_j^{\text{max}}(t) \quad (36)$$

所以问题 P10 可以写成如下形式

$$\begin{aligned} \text{P10: } \min_{f_{ij}(t)} & G_f^{\text{max}} \\ \text{s.t. } \text{C11: } & \sum_i^N x_{ij}(t) \frac{C_2}{G_f^{\text{max}} - \frac{\tau_{i,\text{tran}}(t)}{\tau_i^{\text{tol}}(t)}} = f_j^{\text{max}}(t) \end{aligned} \quad (37)$$

考虑到约束 C11 的等号左边部分是关于  $G_f^{\text{max}}$  单调递减的, 所以可以采用二分搜索迭代法来求解问题 P10。本文提出的基于二分搜索迭代的计算资源分配算法如算法 1 所示。

**算法 1** 基于二分搜索迭代的计算资源分配算法  
设置  $l=1$  和收敛阈值  $\epsilon < 0$ , 设置

$$G_f^{\text{max,min}} = \max \left\{ \frac{\tau_{i,\text{tran}}(t)}{\tau_i^{\text{tol}}(t)} \right\}, G_f^{\text{max,max}} = \sum_i^N \left( \frac{\tau_{i,\text{tran}}(t)}{\tau_i^{\text{tol}}(t)} + \frac{C_2}{f_{ij}(t)} \right)$$

故存在  $G_f^{\text{max,min}} \leq G_f^{\text{max,opt}} \leq G_f^{\text{max,opt}}$ 。

- 1) 循环
- 2)  $G_f^{\text{max,l}} = \frac{G_f^{\text{max,min}} + G_f^{\text{max,max}}}{2}$
- 3) if  $|G_f^{\text{max,max}} - G_f^{\text{max,min}}| \leq \epsilon$  then
- 4)  $G_f^{\text{max,opt}} = G_f^{\text{max,l}}$
- 5) else
- 6) for  $j=1: M$
- 7) if 式(35)不成立 then
- 8)  $G_f^{\text{max,min}} = G_f^{\text{max,l}}$
- 9) else
- 10)  $G_f^{\text{max,max}} = G_f^{\text{max,l}}$
- 11) end if
- 12) end for
- 13) end if
- 14)  $l = l + 1$
- 15) 直到满足收敛阈值。将  $G_f^{\text{max,l}}$  代入式(36),

得到计算资源分配方案  $f_{ij}(t)$ 。

本文所提在线求解 JOSRA 问题的 TOA 算法如算法 2 所示。

**算法 2** TOA 算法

- 1) 定义用户任务  $\text{Task}_i(t)$ , 用户最大容忍响应时延  $\tau_i^{\text{tol}}(t)$ , 资源池总计算资源  $F$ , 服务器承载容量  $Bs_j$ 。
- 2) 定义服务器数  $M$ , 用户数  $N$ , 初始信任值。令时隙数为  $t=1$  和结束时隙为  $T$ 。

- 3) 循环
- 4) 求解问题 P2 得到卸载决策  $\mathbf{X}^*(t)$ 。
- 5) 求解问题 P4 得到最大资源预算分配方案  $f_j^{\max*}(t)$ 。
- 6) 求解问题 P6 得到用户发射功率  $p_i^*(t)$ ，并根据式(3)得到传输时延  $\tau_{i,\text{tran}}(t)$ 。
- 7) 求解问题 P8 得到服务器计算资源分配方案  $f_{ij}^*(t)$ ，并根据式(4)得到计算时延  $\tau_{i,\text{com}}(t)$ 。
- 8) 根据式(14)得到用户满意度  $G_i(t)$ 。
- 9) 根据式(10)和式(12)更新服务器信任值  $\text{rew}_j(t)$  和用户-服务器信任值  $T_{ij}(t)$ 。
- 10)  $t = t + 1$ 。
- 11) 到达结束时隙则结束循环。

### 4 实验仿真

为了验证所提算法的有效性，本文在考虑信任环境下的系统模型中进行在线仿真实验，并与其他算法进行比较后分析 TOA 算法的性能。

#### 4.1 仿真参数设置

在仿真实验中，假设有  $M$  个虚拟服务器、一个 SP 以及实时更新的信任库， $N$  个用户在每个时隙产生不同数据量、不同时延要求的任务。本文假设用户任务全部卸载到服务器进行处理，用户与服务器之间的无线信道增益服从瑞利随机衰落，信道带宽被共用同一服务器的用户平均分配。其余仿真参数如表 1 所示。

表 1 仿真参数设置	
参数	值
用户数量 $N$ / 个	5 ~ 20
服务器数量 $M$ / 个	5
任务数据量 $A_i(t)$ / kbit	5 ~ 20
用户实验敏感度 $\lambda_i(t)$	1 ~ 3
计算资源池 $F$ / GHz	50 ~ 90
服务器最小计算资源 $f^{\min}$ / GHz	10
噪声功率谱密度 $N_0$ / (dBm·Hz <sup>-1</sup> )	-174
服务器计算能力 $S_i$ / (cycle·bit <sup>-1</sup> )	500
用户最大传输功率 $p^{\max}$ / dBm	5
系统带宽 / MHz	20

#### 4.2 对比实验设置

JOSRA 主要包括任务卸载和资源分配 2 个阶段，故考虑将是否在 2 个阶段考虑信任以及仅考虑

满足最大容忍时延的 3 种算法作为对比实验，来证明本文所提 TOA 算法的有效性。对比算法如下。

1) 文献[22]提出的效用感知计算卸载算法，针对本文的研究问题进行调整并简称为 AUTO。AUTO 的目标是最大化服务器的资源利用率，将较少的计算资源用于任务处理。因此，每个用户任务将在其最大容忍时延下完成。AUTO 下所有的用户满意度  $G = 1$ 。

2) 贪婪卸载和基于信任的资源分配，简称为 GOTA，用户在任务卸载阶段贪婪地选择计算能力强的服务器进行卸载，保证计算能力强的服务器为满负载状态，不考虑信任的影响；在资源分配阶段则与本文所提算法一致，考虑信任可能引起的夸大资源报价后，再对计算资源进行分配。

3) 基于信任的任务卸载和随机资源分配，简称为 TORA，在任务卸载阶段，基于信任完成卸载策略；在资源分配阶段，不考虑信任折扣带来的“欺诈”行为，即  $f_{ij}^{\text{real}}(t) = f_{ij}(t)$ 。随机分配服务器上的计算资源，且要求保证服务器计算资源效用为 1。

#### 4.3 仿真结果

针对 JOSRA 模型，本文提出的 TOA 算法能够实现高满意度低时延且显著优于对比算法。在线仿真 500 个时隙，服务器的实时信任值如图 2 所示。从图 2 可以看出，在信任建立初期的前 50 个时隙，信任值开始趋于稳定。在  $N = 10, M = 5, F = 70$  的仿真环境下，用户-服务器的信任值如图 3 所示。从图 3 可以看出，用户对不同的服务器信任值不同。当选择最信任的服务器进行卸载时，都能获得高于 0.8 的信任折扣等级，有效保证了用户的服务质量。

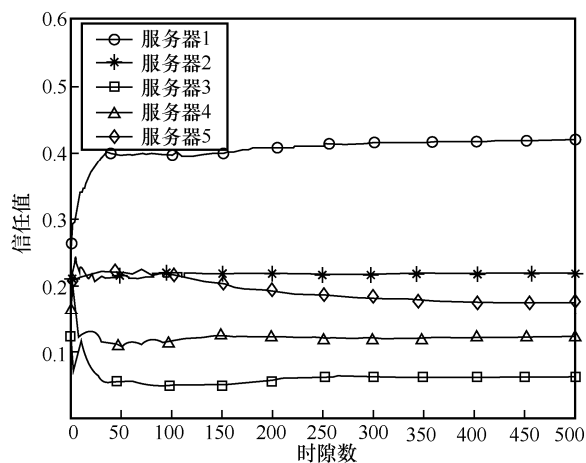


图 2 服务器的实时信任值

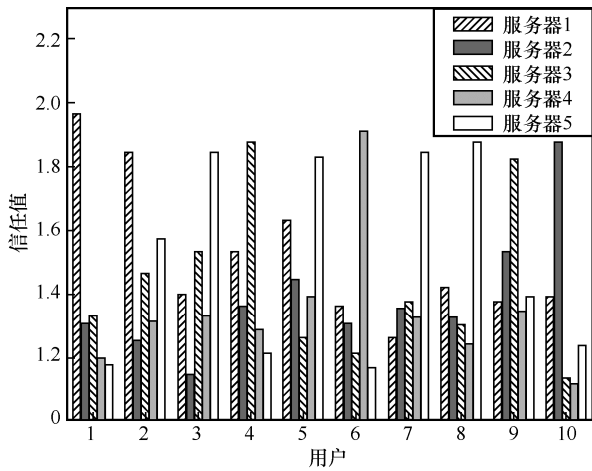
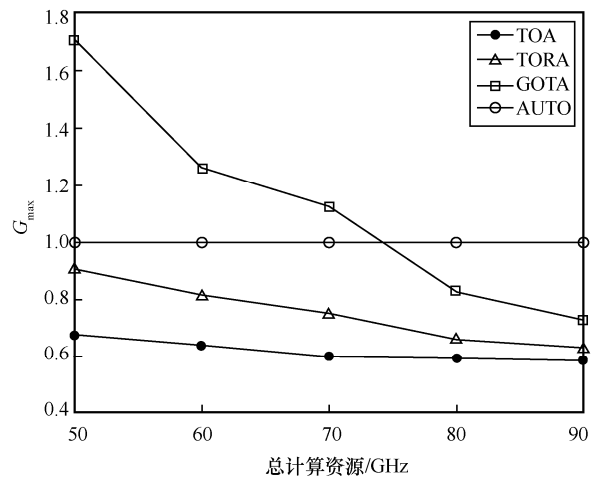


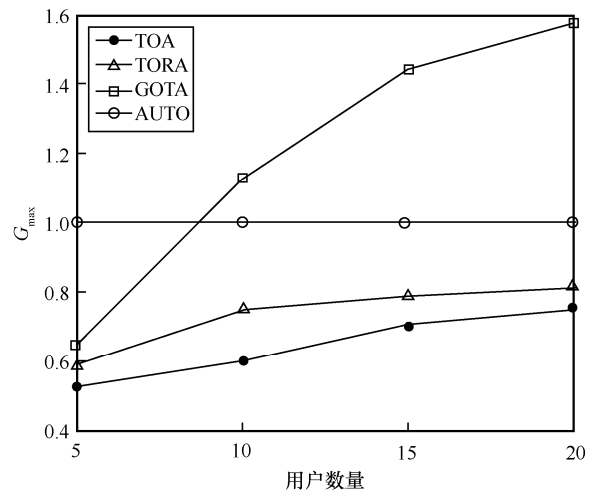
图 3 用户-服务器的信任值

为了观察长期的系统公平性，本文以 450 时隙之后稳定信任状态下不同算法的数据为基础，获得系统的最差用户满意度  $G_{\max}$ 、用户平均满意度  $G_{\text{ave}}$  和平均响应时延  $\tau$  并进行分析论证。

1) 最差用户满意度  $G_{\max}$ 。首先，分析不同算法的最差用户满意度，如图 4 所示。在图 4(a)中，除 AUTO 外（因为该算法令任务处理时延等于最大容忍时延，即  $G_{\max} = 1$ ）， $G_{\max}$  值都随着总计算资源的增加而减小，因为随着资源池的资源增加，每个服务器分到的资源预算也增加，使用户可以获得更多的计算资源，缩短任务完成的响应时间，从而提高用户满意度。在计算资源紧张的情况下，对比算法 GOTA 存在  $G_{\max}$  值大于 1 的情况，意味着平均每个时隙内至少有一个用户不能在最大容忍时延内完成任务响应，这是由贪婪卸载策略引起的，因为它忽视了用户-服务器之间的信任关系，服务器不会提供足够计算资源给不信任的用户。所提算法 TOA 和对比算法 TORA 均能保证  $G_{\max}$  值小于 1，即满意度最差的用户也能保证在最大容忍时延内完成任务响应，但总体而言，所提算法 TOA 的  $G_{\max}$  值最小，TORA、AUTO 和 GOTA 分别比 TOA 高 21.8%、61.6% 和 82.9%，最有效地保证满意度最差的用户体验。图 4(b)中展示了用户数量对  $G_{\max}$  值的影响，除 AUTO 外，其余算法的  $G_{\max}$  值都随着用户数量的增多而增大。这时因为总计算资源不变，随着用户数量的增加，用户对共享资源的竞争将更加激烈，对满意度最差的用户资源倾斜十分有限，从而使  $G_{\max}$  值变大。但本文所提 TOA 仍能保证最差用户的响应时延远小于最大容忍时延且优于其他算法，有效保障了用户体验。



(a) 不同总计算资源下的最差用户满意度

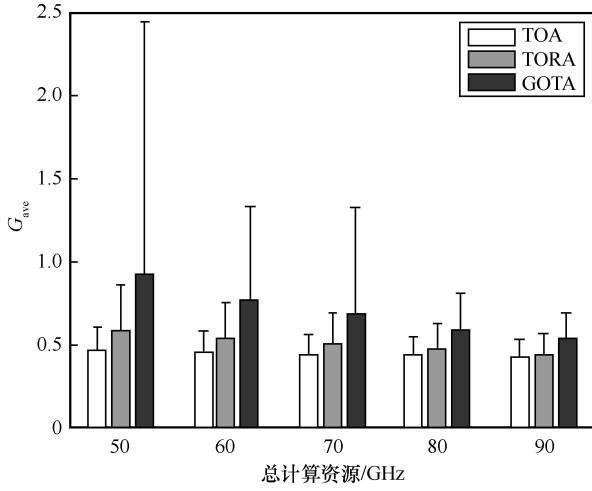


(b) 不同用户数量下的最差用户满意度

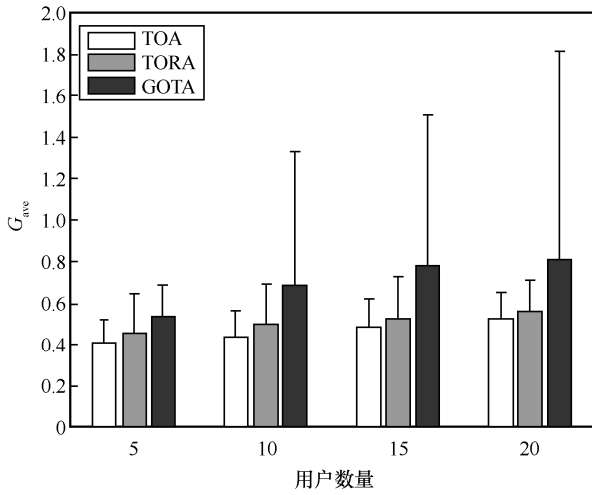
图 4 不同算法的最差用户满意度

2) 用户平均满意度  $G_{\text{ave}}$ 。不同算法的用户平均满意度如图 5 所示，考虑 AUTO 算法下的  $G$  值恒为 1，故不再展示。其中，柱状图表示所有用户在长期表现下  $G_{\text{ave}}$  的均值，误差棒则是用户所有时隙下  $G_{\text{ave}}$  的标准差。从图 5(a)和图 5(b)中可以看出，不论是改变总计算资源还是改变用户数量，所提算法的  $G_{\text{ave}}$  的均值和标准差的变化趋势基本一致且都显著低于其他 2 种算法。在总计算资源充足和用户数量较少的情况下，3 种算法均有较好的表现。但随着总计算资源的减少或是用户数量的增多，TORA 和 GOTA 算法下  $G_{\text{ave}}$  的均值和标准差都会出现较大增长，表明在资源紧张的情况下，系统不能相对公平地兼顾每一个用户，故用户对服务的满意度表现参差不齐。而本文所提 TOA 算法在资源受限情况下仍能保持良好的用户平均满意度，当总计算资源为 50 GHz 时，TOA 算法下的  $G_{\text{ave}}$  均值为

0.473, 分别优于 TORA 和 GOTA 24.3%和 95.3%; 当用户数量为 20 时, TOA 算法下的  $G_{ave}$  均值为 0.526, 分别优于 TORA 和 GOTA 6.9%和 55.1%。



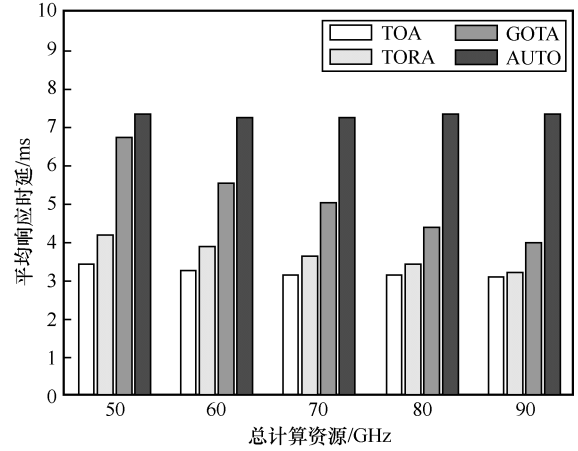
(a) 不同总计算资源下的用户平均满意度



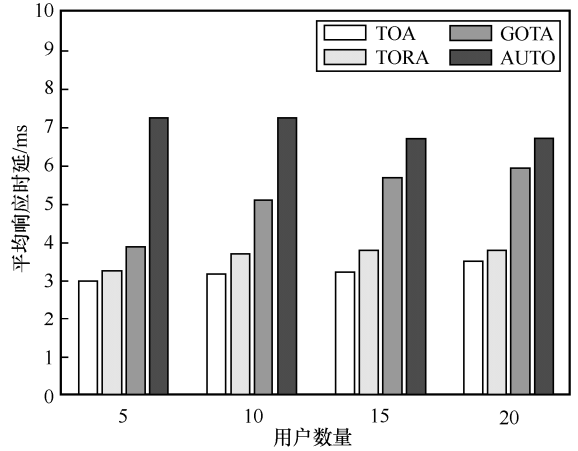
(b) 不同用户数量下的用户平均满意度

图 5 不同算法的用户平均满意度

3) 平均响应时延  $\tau$ 。图 6 展示了不同算法的用户平均响应时延。AUTO 的目标是满足用户的最小需求, 所以响应时延等于最大容忍时延。GOTA 在卸载决策时缺少对信任值的考虑, 使用户不能获得服务器的高效服务, 平均总时延远高于 TOA 下的总时延。在计算资源紧张的情况下, GOTA 的不足表现更加明显。TORA 尽管在资源分配时没有考虑信任折扣的问题, 但在匹配阶段是基于信任完成卸载决策的, 所以性能优于 GOTA 和 AUTO。而本文所提 TOA 能在保证用户满意度稳定的同时使用户平均响应时延最小, 即在保证系统公平性的同时使用户获得良好的用户体验。



(a) 不同总计算资源下的平均响应时延



(b) 不同用户数量下的平均响应时延

图 6 不同算法的用户平均响应时延

### 5 通信计算资源的联合优化分析

在处理问题 P5 时, 本文采用将优化整个过程的公平性硬解耦为优化通信和计算 2 个部分的公平性的方式。这样做虽然降低了算法复杂度, 但也忽略了两者的耦合性, 牺牲了求解精度。本节通过迭代  $p_i(t)$  和  $f_{ij}(t)$  的方式直接优化整个过程的公平性, 在单个时隙内进行仿真, 并分析 2 种方法的差异。

在迭代过程中, 对  $p_i(t)$  变量优化时需要考虑  $f_{ij}(t)$  的影响, 所以优化问题 P6 需改写成 P6'。

$$\begin{aligned}
 \text{P6'} : \min_{p_i(t)} \max_{i \in \mathcal{N}} & \frac{\text{lb} \left( 1 + \frac{p_i^{\min} h_{ij}}{N_0} \right)}{\lambda_i(t) \text{lb} \left( 1 + \frac{p_i(t) h_{ij}}{N_0} \right)} + \frac{\tau_{i,\text{com}}(t)}{\tau_{i,\text{com}}^{\text{tol}}(t)} \\
 \text{s.t. C6} & \hspace{15em} (38)
 \end{aligned}$$

与 P6 相比, 目标函数增加了计算部分的满意度。通信计算资源的迭代优化算法如算法 3 所示。

**算法3** 通信计算资源的迭代优化算法

初始化系统参数，设置初始值  $p_i^{(0)}(t)$ ,  $f_{ij}^{(0)}(t)$  和最大迭代次数  $L^{\max}$ 。

- 1) while  $l \leq L^{\max}$  do
- 2) 设置迭代次数  $l = l + 1$
- 3) 固定  $f_{ij}^{(l-1)}(t)$ ，使用凸优化工具箱求解 P6' 计算  $p_i^{(l)}(t)$ 。
- 4) 固定  $p_i^{(l)}(t)$ ，根据算法 1 求解 P8 计算  $f_{ij}^{(l)}(t)$ 。
- 5) 将  $p_i^{(l)}(t)$  和  $f_{ij}^{(l)}(t)$  代入式(28)，更新最差用户满意度  $G_{\max}^{(l)}$ 。
- 6) end while

使用算法 3 在时隙  $t = 500$  下迭代优化求解不同用户数量、不同计算资源下的问题 P5，迭代优化与解耦优化对比如图 7 所示。用户数为 10、计算资源为 70 的情况下最差用户满意度在 2 次迭代优化后收敛至 0.498，而解耦优化得到最差用户满意度为 0.542，当用户数为 5、计算资源为 90 时，2 种优化方法最终得到的最差用户满意度为 0.371 和 0.428。虽然解耦优化精度略低于迭代优化方法，但不需要进行迭代运算，复杂度低。

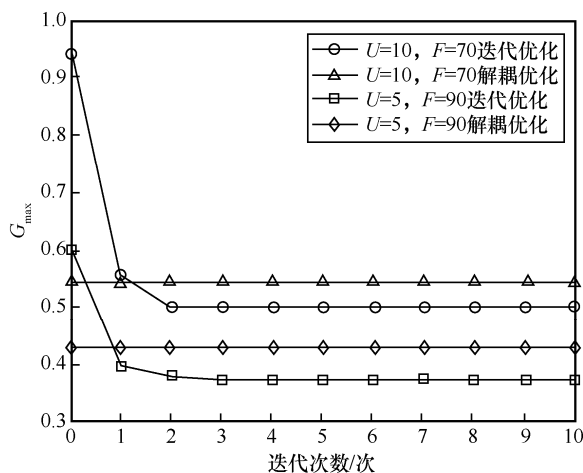


图7 迭代优化与解耦优化对比

## 6 结束语

本文研究信任环境下基于边缘计算的卸载决策和资源分配策略，建立实时信任模型并利用 min-max 理论保障系统的公平性。在每个时隙内，通过联合优化卸载决策、资源池分配、通信资源和计算资源，提出优化最差用户满意度来保证整个系

统的公平性的问题，提出 TOA 算法，基于回溯算法确定卸载策略，采用拉格朗日乘子法、凸优化工具箱和二分迭代搜索法完成对资源配置的优化，最终实现了系统公平性且保障了用户的满意度。

## 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] CAI Q, ZHOU Y Q, LIU L, et al. Collaboration of heterogeneous edge computing paradigms: how to fill the gap between theory and practice[J]. IEEE Wireless Communications, 2024, 31(1): 110-117.
- [3] CUI Y H, LIU F, JING X J, et al. Integrating sensing and communications for ubiquitous IoT: applications, trends, and challenges[J]. IEEE Network, 2021, 35(5): 158-167.
- [4] ABBAS N, ZHANG Y, TAHERKORDI A, et al. Mobile edge computing: a survey[J]. IEEE Internet of Things Journal, 2018, 5(1): 450-465.
- [5] 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.
- [6] YAN L, YANG K, YANG S Y, et al. A trust-aware fog offloading game with long-term trustworthiness of users[C]//Proceedings of the 2021 IEEE Global Communications Conference. Piscataway: IEEE Press, 2021: 1-6.
- [7] 齐平, 王福成, 徐佳, 等. 移动边缘计算环境下基于信任模型的可靠多重计算卸载策略[J]. 计算机集成制造系统, 2020, 26(6): 1616-1627.
- [8] 元晋, 孙海蓉, 巩锟, 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.
- [9] 王妍, 韩笑, 曾辉, 等. 边缘计算环境下服务质量可信的任务迁移节点选择[J]. 计算机科学, 2020, 47(10): 240-246.
- [10] WANG Y, HAN X, ZENG H, et al. Task migration node selection with reliable service quality in edge computing environment[J]. Computer Science, 2020, 47(10): 240-246.
- [11] WU D X, SHEN G H, HUANG Z Q, et al. A trust-aware task offloading framework in mobile edge computing[J]. IEEE Access, 2019, 7: 150105-150119.
- [12] 熊小峰, 黄淳岚, 乐光学, 等. 边缘计算中基于综合信任评价的任务卸载策略[J]. 电子学报, 2022, 50(9): 2134-2145.
- [13] XIONG X F, HUANG C L, LE G X, et al. Task offloading scheme based on comprehensive trust evaluation in edge computing[J]. Acta Electronica Sinica, 2022, 50(9): 2134-2145.
- [14] GUO C T, HE W, LI G Y. Optimal fairness-aware resource supply and

- demand management for mobile edge computing[J]. IEEE Wireless Communications Letters, 2021, 10(3): 678-682.
- [13] GUO K, ZHANG R L. Fairness-oriented computation offloading for cloud-assisted edge computing[J]. Future Generation Computer Systems, 2022, 128:132-141.
- [14] ZHANG L P, CHAI R, YANG T T, et al. Min-max worst-case design for computation offloading in multi-user MEC system[C]//Proceedings of the IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops. Piscataway: IEEE Press, 2020: 1075-1080.
- [15] LAI S, FAN X C, YE Q W, et al. FairEdge: a fairness-oriented task offloading scheme for IoT applications in mobile cloudlet networks[J]. IEEE Access, 2020, 8: 13516-13526.
- [16] LIN X, WU J, BASHIR A K, et al. FairHealth: long-term proportional fairness-driven 5G edge healthcare in Internet of medical things[J]. IEEE Transactions on Industrial Informatics, 2022, 18(12): 8905-8915.
- [17] CARONNI G. Walking the Web of trust[C]//Proceedings of IEEE 9th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises. Piscataway: IEEE Press, 2002: 153-158.
- [18] 张琳, 魏新艳, 刘茜萍, 等. 基于协作信誉和设备反馈的物联网边缘服务器信任评估算法[J]. 通信学报, 2022, 43(2): 118-130.
- ZHANG L, WEI X Y, LIU X P, et al. Trust evaluation algorithm of IoT edge server based on cooperation reputation and device feedback[J]. Journal on Communications, 2022, 43(2): 118-130.
- [19] AHMED A, ABU B K, CHANNA M I, et al. Energy-aware and secure routing with trust for disaster response wireless sensor network[J]. Peer-to-Peer Networking and Applications, 2017, 10(1): 216-237.
- [20] MITSIS G, APOSTOLOPOULOS P A, TSIROPOULOU E E, et al. Intelligent dynamic data offloading in a competitive mobile edge computing market[J]. Future Internet, 2019, 11(5): 118.
- [21] ZHOU J Y, ZHANG X L. Fairness-aware task offloading and resource

allocation in cooperative mobile-edge computing[J]. IEEE Internet of Things Journal, 2022, 9(5): 3812-3824.

- [22] XU X L, HE C X, XU Z Y, et al. Joint optimization of offloading utility and privacy for edge computing enabled IoT[J]. IEEE Internet of Things Journal, 2020, 7(4): 2622-2629.

#### [作者简介]



杨守义（1965—），男，河南民权人，博士，郑州大学教授、博士生导师，主要研究方向为无线移动通信、移动云计算等。



李富康（1998—），男，河南民权人，郑州大学硕士生，主要研究方向为移动边缘计算、无线通信等。



任瑞敏（2000—），女，河南周口人，郑州大学硕士生，主要研究方向为移动边缘计算和虚拟现实技术等。