

基于移动边缘计算环境下的服务缓存 和任务调度联合优化算法

杜建华^{1,3}, 王立俊^{1,3}, 谢寒生¹, 赵卓宁², 王双双¹

(1. 海南省气象信息中心, 海口 570203; 2. 成都信息工程大学, 成都 610225;

3. 海南省南海气象防灾减灾重点实验室, 海口 570203)

摘要: 在移动边缘计算中, 在资源有限的边缘设备上对服务缓存和任务执行进行合理的决策能够大幅度地提高卸载效率和减少应用程序的处理时延; 针对边缘计算环境下服务缓存与任务卸载决策问题, 建立网络模型和服务缓存模型, 定义关联的边缘设备、协作的边缘设备任务执行时延及远端云数据传输和任务执行的时延, 提出一种联合优化算法来求解任务执行时延约束条件下的服务缓存决策最优解; 该算法采用粒子群优化 Particle Swarm Optimization (PSO), 将移动用户的任务按照整数编码, 优化任务处理时延适应度和粒子速度更新, 缩短任务延迟时间; 仿真实验结果表明, 联合优化算法取得相比其他策略完成时间更少且能适应大规模任务调度的效果。

关键词: 移动边缘计算; 服务缓存; 任务卸载; 资源调度; 效率

Joint Optimization Algorithm of Service Cache and Task Scheduling Based on Mobile Edge Computing Environment

DU JianHua^{1,3}, WANG Lijun^{1,3}, XIE Hansheng¹, ZHAO Zhuoning², WANG Shuangshuang¹

(1. Meteorological Information Center of Hainan Province, Haikou 570203, China;

2. Chengdu University of Information Technology, Chengdu 610225, China;

3. Key Laboratory of South China Sea Meteorological Disaster Prevention and Mitigation of Hainan Province, Haikou 570203, China)

Abstract: Making reasonable decisions for service cache and task execution on edge devices with limited resources in the mobile edge computing can greatly improve the unloading efficiency and reduce processing delay of applications. With defining the task execution delay of associated edge devices, cooperative edge devices and remote cloud, a joint optimization algorithm was proposed by establishing a network model and service cache model to solve the optimal solution of service caching decision under the constraint of task execution delay on the problem of service cache and task unloading decision. The algorithm with PSO used integer encoding of mobile users' tasks, optimization of task processing delay fitness and particle velocity updating to shorten the task delay time. The simulation results showed that the algorithm can obtain less completion time in the large-scale task scheduling than other strategies.

Keywords: Mobile Edge Computing; Service Cache; Task Unloading; Resource scheduling; Efficiency

0 引言

物联网以及无线通信技术的推陈出新, 使得各种移动相关应用呈爆炸式发展势头。大量的计算密集和时间敏感型应用, 在通信资源和计算资源方面对网络基础设施提出了更高的要求^[1-4]。而传统云计算框架中, 网络带宽因终端设备与云中心服务器之间要处理大量数据上传、处理和结果的回传, 承受巨大的压力。为提高资源效率和用户体验, 移动边缘计算 (MEC, mobile edge computing) 框架能够给予移动终端以任务卸载支持, 使得云计算服务从集中云扩展至网络边

缘, 在近端提供资源的同时有效降低了服务延迟^[5-9]。需要注意的是, 边缘设备在通信、计算、存储资源等方面具有一定的局限性, 尤其是当终端用户的任务需求激增时。大量终端用户向边缘设备卸载任务, 会造成边缘设备资源紧张、负载过重和任务处理时延的增加, 影响任务处理的时效和用户体验。此外, 各边缘设备间的负载不均衡现象会进一步加剧, 部分闲置的服务资源无法被充分利用。

为了给终端用户提供更准确、更精确粒度的服务, 一些基于移动边缘计算的服务缓存策略被提出。文献 [10]

收稿日期: 2022-02-17; 修回日期: 2022-03-17。

基金项目: 国家自然科学基金(41775011); 海南省气象局科技创新项目(HNQXSJ202118)。

作者简介: 杜建华(1981-), 男, 湖北洪湖人, 硕士, 高级工程师, 主要从事气象数据理论和应用研究。

通讯作者: 王立俊(1989-), 男, 海南海口人, 硕士, 工程师, 主要从事气象数据处理与应用的研究。

引用格式: 杜建华, 王立俊, 谢寒生, 等. 基于移动边缘计算环境下的服务缓存和任务调度联合优化算法[J]. 计算机测量与控制, 2022, 30(5): 238-242, 250.

对支持 MEC 的密集蜂窝网络中的动态服务缓存策略进行了研究, 提出了一种高效的基于联合优化动态服务缓存和任务卸载的在线算法。文献 [11] 针对 MEC 系统在服务异构性、空间需求耦合和分散协调等方面存在的问题, 基于 MEC 的密集小区网络中的协作服务部署方法进行了分析, 提出了一种高效的分布式算法。文献 [12] 研究了具有多维约束的服务部署和请求路由的联合优化问题, 提出了一种使用随机取整实现接近最优性能的算法。为实现更可行的边缘资源分配, 文献 [13] 提出了齐次条件下的常数因子近似算法和一般条件下的启发式算法来解决缓存服务放置和任务请求调度问题。文献 [14] 引入卸载成本模型来捕获用户能耗、服务缓存成本和云使用成本, 利用局部搜索技术设计了一个多项式时间迭代算法 COSTA。为了充分利用边缘节点的存储和计算能力, 文献 [15] 针对边缘节点之间的协作服务缓存和工作负载调度问题建立数学模型, 并设计了迭代算法来解决上述混合整数非线性规划问题。从上述研究工作可以看到, 边缘设备之间的协作能够更高效地利用各设备的通信、计算和存储资源。计算任务卸载决策需要对众多终端用户的服务响应进行合理的安排, 同时尽可能地避免边缘设备间的负载不均衡的情况。任务卸载和服务缓存之间相互耦合, 进行联合优化能够在任务执行时延限制的条件下, 有效保障用户的服务质量。

1 问题描述

1.1 网络模型

考虑如图 1 所示的移动边缘计算网络, K 个基站配备了具有计算和存储功能的边缘服务器, N 个移动用户产生任务后, 指定服务器或远程云数据中心来执行。考虑到边缘服务器的密集部署以及覆盖范围会出现重叠, 对于移动用户 n 可能位于多个不同基站的控制区域, 用 K_n 表示该用户所能访问的基站集合。由于任务的执行需要用户的输入及计算资源, C_k 和 F_k 分别表示基站 k 的最大容量和最大周期频率。系统能够处理的任务种类为 S , 且任意移动用户在一个时间段提交一个任务请求。 $O_{n,s}$ 表示用户 n 的任务请求 s , 计算任务模型用三元组 $O_{n,s} = \{f_{n,s}, d_{n,s}, T_{n,s}\}$ 来描述, 其中, $f_{n,s}$ 为任务所需的计算资源, $d_{n,s}$ 为任务的输入数据量, $T_{n,s}$ 为任务所允许的时间延迟。

当移动用户的服务请求被迁移到邻近的基站组上, 如果该基站已缓存移动用户请求的服务并具有足够的计算能力和带宽资源以供移动用户使用, 则直接进行处理。如果邻近的基站组中没有缓存所请求的服务, 则接收到服务请求的边缘设备将请求其他缓存相应的服务的边缘设备进行协同工作, 该移动用户的服务请求将被迁移至协助处理的边缘设备。若所有的边缘设备都无法进行响应, 则直接将任务上传至远端中心云来进行处理。

1.2 服务缓存模型

移动用户卸载的计算任务与边缘设备缓存的服务相关联, 任务能否得以执行取决于边缘设备上是否缓存了对应

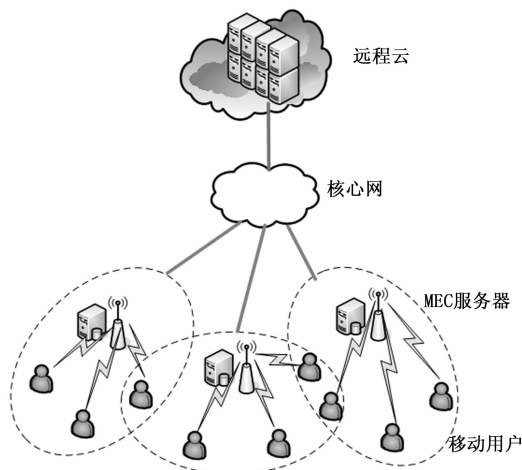


图 1 网络模型

的服务。由于边缘设备上的缓存容量有限, 无法缓存全部的服务。因此, 边缘设备需要定期地与覆盖范围内的移动用户进行交互, 以确定将哪些服务置于其缓存中。对于时隙 t , 用 $\Theta_t = \{\Theta_{k,t} \mid k \in K\}$ 表示时隙 t 所有边缘设备的服务缓存策略, 其中: $\Theta_{k,t} = \{\Theta_{k,t}(s) \mid s \in S\}$ 为为时隙 t 边缘设备 k 中所缓存的服务集合。 $\Theta_{k,t}(s)$ 是一个二元变量, 取值为 1 表示边缘设备 k 在时隙 t 时缓存了服务 s , 取值为 0 则表示未缓存服务 s 。边缘设备所缓存的服务集合不能超过其缓存容量上限, 因此, 服务缓存约束条件可以表示为:

$$\sum_{s \in S} \theta_{k,t}(s) \leq C_k, \forall k \in K \quad (1)$$

1.3 任务卸载策略

相比于移动用户, 边缘设备的计算、缓存等资源比较充足, 移动用户将任务卸载至相关联的边缘设备进行处理, 能有效弥补移动终端资源不足、能耗受限等问题。任务卸载处理的前提条件是边缘设备是否对相应的服务事先予以缓存。由于边缘设备资源有限, 若因缓存不足没有对用户请求的服务进行缓存, 它会先寻求其他缓存相关服务的边缘设备进行协作处理。如果其他边缘设备也无法处理, 则再将计算任务卸载至远端云服务中心处。因此, 移动用户产生的计算任务的卸载优先顺序依次为: 关联的边缘设备、协作的边缘设备以及远端云服务中心。边缘设备间通过核心网互联, 边缘设备间的传输速率可以看成是定值, 用 R_0 表示。若移动用户的任务请求无法通过边缘设备处理, 而需要在远端云服务中心处执行时, 将由关联的边缘设备通过核心网和回程链路传输至远端, 用恒定值 t_c 来表示这部分的传输时延。另外, 由于处理后的数据量一般远小于任务的输入, 且移动用户端的下载速率远高于其上传速率, 本文不考虑边缘设备或云数据中心处理后的数据下载延时。

假设移动用户上传任务数据为恒定功率 P , 时隙 t 内的信道状态为 $H(t) = \{h_{k,n}(t) \mid k \in K, n \in N\}$, 其中 $h_{k,n}(t)$ 表示移动用户 n 到边缘服务器 k 之间的信道增益。 $B(t) = \{b_{s,n}(t) \mid s \in S, n \in N\}$ 表示该时隙带宽资源分

配, $b_{s,n}(t) \in [0, 1]$ 表示接入边缘设备为卸载移动用户 n 的 s 类任务所分配的带宽资源比例。考虑到边缘设备带宽资源限制, $b_{s,n}(t)$ 需满足约束条件:

$$\sum_{n \in N} \sum_{s \in S} \delta_{s,n}(t, k) b_{s,n} \leq 1, \forall k \in K \quad (2)$$

其中: $\delta_{s,n}(t, k)$ 为二元变量, 取值 1 表示在时隙 t 中移动用户 n 的任务类型 s 将被卸载到边缘服务器 k 进行处理, 反之取值为 0。

根据接入控制和带宽资源分配决策, 移动用户 n 在时隙 t 用于任务卸载时的数据传输速率可表示为:

$$R_{s,n}(t) = \sum_{k \in K_n} \delta_{s,n}(t, k) b_{s,n}(t) \frac{B_k}{|M_k|} \log_2 \left(1 + \frac{P h_{k,n}(t)}{N_0} \right) \quad (3)$$

其中: N_0 为噪声功率谱密度, B_k 表示边缘设备 k 的信道带宽, $|M_k|$ 表示边缘设备 k 关联的移动用户个数。

1.3.1 关联的边缘设备执行任务

若移动设备产生的任务卸载到自身关联的边缘设备处执行, 则执行时延包括: 移动用户 n 上传数据到边缘设备所需的传输时延以及边缘设备执行任务所需的计算时延。

$$t_{n,s}(k) = \frac{d_{n,s}}{R_{s,n}(t)} + \frac{f_{n,s}}{F_{k,n}}, \sum_{n \in N} F_{k,n} \leq F_k \quad (4)$$

其中: $F_{k,n}$ 表示边缘设备 k 分配给任务 $\mathcal{O}_{n,s}$ 的计算量。

1.3.2 协作的边缘设备执行任务

若移动设备相关联的边缘设备处没有缓存任务相应的服务, 边缘设备会寻求其他能够予以协作的边缘设备。这种情况下, 时间延迟包括: 移动用户上传数据到相邻边缘设备所需的传输时延, 边缘设备之间传输数据所需的传输时延以及进行协作的边缘设备执行任务所需的计算时延。

$$t_{n,s}(k, l) = d_{n,s} \left(\frac{1}{R_{s,n}(t)} + \frac{dis(k, l)}{R_0} \right) + \frac{f_{n,s}}{F_{l,n}}, \sum_{n \in N} F_{l,n} \leq F_k \quad (5)$$

其中: $dis(k, l)$ 表示边缘设备 k 和 l 之间的链路长度, $F_{l,n}$ 表示边缘设备 l 分配给任务 $\mathcal{O}_{n,s}$ 的计算量。

1.3.3 远端云执行任务

若关联的边缘设备和其他边缘设备都没有缓存移动用户卸载的计算任务所对应的服务, 关联的边缘设备需要将任务发送至远端云服务中心处执行。考虑到远端云计算和存储资源丰富, 故不考虑其计算所需时间开销。此时所需的时延包括: 移动用户至边缘设备间的数据传输时延以及边缘设备至远端云服务中心间的数据传输时延。因此, 任务被处理所需要的时间开销为:

$$t_{n,s}(k, z) = \frac{d_{n,s}}{R_{s,n}(t)} + t_z \quad (6)$$

1.4 目标函数

每个移动用户的计算任务选择合适的关联边缘设备作为任务卸载目标, 关联的边缘设备若提前缓存了该任务执行所需的服务则分配计算资源予以执行, 否则将向其他边缘设备提出协作请求或发送至远端云服务中心进行处理。本文目标为通过对任务卸载与缓存决策进行合理调度, 优

化通信和计算资源分配, 实现所有移动用户的任务的平均处理时间延迟的最小化。因此, 优化目标表示为:

$$\frac{1}{N} \sum_{n \in N} T_n =$$

$$\frac{1}{N} \sum_{n \in N} [t_{n,s}(k) \quad t_{n,s}(k, l) \quad t_{n,s}(k, z)] \cdot [\omega_\alpha \quad \omega_\beta \quad \omega_\gamma]^T \quad (7)$$

$$\sum_{s \in S} \theta_{k,s}(s) \leq C_k, \forall k \in K \quad (8)$$

$$\omega_\alpha + \omega_\beta + \omega_\gamma = 1, \forall \omega_\alpha, \omega_\beta, \omega_\gamma \in \{0, 1\} \quad (9)$$

$$F_{k,n} > 0, \forall n \in N, \forall k \in K \quad (10)$$

$$\sum_{n \in N} F_{k,n} \leq F_k, \exists \omega_\alpha = 1, \forall k \in K \quad (11)$$

$$\sum_{n \in N} F(l, n) \leq F_l, \exists \omega_\beta = 1, \forall l \in K \quad (12)$$

$$[t_{n,s}(k) \quad t_{n,s}(k, l) \quad t_{n,s}(k, z)] \cdot [\omega_\alpha \quad \omega_\beta \quad \omega_\gamma]^T \leq t_{n,s}, \forall k \in K \quad (13)$$

式 (8) 的约束条件为边缘设备缓存的服务不能超过其缓存容量。式 (9) 的约束条件为移动用户的计算任务的可以为关联的边缘设备、协作的边缘设备或远端云服务中心。式 (10) 表示边缘设备为移动用户的计算任务分配的计算量为正。式 (11) 表示边缘设备为用户任务分配的计算资源不能超过其计算资源总量。式 (12) 表示协作的边缘设备同样为用户分配的计算资源不能超过其资源总量限制。式 (13) 约束条件为所有任务的处理时间需满足其时延限制。

2 算法设计

上述任务卸载与缓存决策调度问题本质上为多目标组合优化问题。算法目标是在满足边缘设备的缓存容量、计算资源分配的约束条件下, 将移动用户的任务请求分配至关联的边缘设备, 使得任务处理的耗费时间最少, 实现对用户的及时响应。对于此类问题, 启发式算法能够在一定的范围内求解次优解或以一定的概率求其最优解, 其中, 粒子群优化 (PSO, particle swarm optimization) 算法具有参数设置少、全局搜索能力强的优点, 其并行性和分布式的特点适合用来求解上述任务卸载与缓存决策调度问题^[16-18]。首先, 需要对粒子进行编码, 使缓存策略、任务卸载与粒子的位置、速度等联系起来。

2.1 粒子编码

假设粒子群规模为 U , 所有移动用户的任务集合 $\Phi = \{\mathcal{O}_1, \mathcal{O}_2, \dots, \mathcal{O}_N\}$ 。粒子个体的编码维度与任务集合元素数量一致, 并采用整数编码。每个粒子元素的取值范围为 $[0, K]$, 其中 0 表示移动用户的任务在远端云服务中心被处理, $1 \sim K$ 则表示某移动用户的任务卸载至相应编号的边缘设备进行处理。每个粒子的卸载决策向量 $X = \{x_1, x_2, \dots, x_N\}$ 表示所有任务最优的执行位置。初始化时, x_i 在 0 到 K 之间随机取值, 且需要保证其对应的服务请求和需要的计算能力满足对应的边缘设备的资源限制条件。粒子的速度为调整当前任务分配至其他边缘设备的趋势快慢, 用 $V = \{v_1, v_2, \dots, v_N\}$ 来表示。 $pBest_i = \{p_{i1}, p_{i2},$

..., p_i, N } 为第 i 个粒子的个体极值, $pBest = \{p_{g1}, p_{g2}, \dots, p_{gN}\}$ 为全局最优粒子。

2.2 适应度函数

算法的优化目标为最优的适应度。本文建立的模型优化目标是在满足边缘设备的缓存容量、计算资源分配的约束条件下, 将移动用户的任务请求分配至关联的边缘设备, 使得任务处理的耗费时间最少。因此, 可以将适应度函数定义为:

$$f = \frac{1}{1 + \exp\left(-\frac{1}{N} \sum_{n \in N} T_n\right)} \quad (14)$$

可见, 所有移动用户的任务的平均处理时间延迟值越大, 优化性能越差; 反之则优化性能越强。

2.3 反向粒子的选择

粒子群优化在算法迭代到一定程度后, 会因为种群多样性的逐渐收窄而可能陷入局部最优^[19-21]。为了提高求解精度和获得比较好的稳定性, 本文将反向学习 (OBL, opposition-based learning) 的理论引入对算法进行改进。其主要思想为: 在求解过程中, 根据每个粒子个体对应的反向粒子, 将其假定为可能得到更接近最优解的粒子个体, 通过粒子个体及其反向个体来提高种群的多样性, 再从中择优选择作为后续迭代的条件。

1) 反向点的选择。将 $\{x_1, x_2, \dots, x_N\}$ 看做是 N 维空间上的一个点, 则其对应的反向点的为 $\{x'_1, x'_2, \dots, x'_N\}$, 有:

$$x'_i = \max_{1 \leq j \leq N} \{x_j\} + \min_{1 \leq j \leq N} \{x_j\} - x_i \quad (15)$$

2) 基于反向解的优化。由于上述优化目标函数为求解任务卸载与缓存决策调度问题的最大值, 若 $f(x'_i) > f(x_i)$ 成立, 则用 x'_i 替换 x_i 。

2.4 速度更新

考虑到粒子一旦陷入局部极值, 则其速度无法进行更新, 且过低的速度会造成粒子不易从局部极值范围内脱离。本文采取极值扰动的方法拓展粒子的搜索区间, 对学习因子采用非线性变化的策略来调整粒子的社会学习和自我学习能力。粒子的速度更新公式修改为:

$$\begin{cases} v_m(u+1) = \\ v_m(u) + c_1(p_m(u) - x_m(u)) + c_2(p_{gm}(u) - x_m(u)) \\ x_m(u+1) = x_m(u)(1 + \psi(\sigma)) + v_m(u+1) \end{cases} \quad (16)$$

$c_1 = 1.3 + 1.2 \cos(\pi u / u_{max})$, $c_2 = 2.0 - 1.2 \cos(\pi u / u_{max})$, c_1, c_2 为学习因子, $\psi(\cdot)$ 为高斯分布函数, δ 取搜索空间长度的 0.2 倍, u_{max} 为最大迭代次数。

3 实验结果与分析

为了验证本文提出的移动边缘计算环境下的数据缓存和任务调度联合优化模型的性能, 采用 CloudSim 进行了仿真实验。实验参数设置为: 终端用户的传输功率为 0.5 W, 每个边缘服务器的信道带宽、存储容量和计算容量分别被设置为 10 MHz、200 GB 和 25 GHz, 噪声功率 100 dBm,

服务类型总数为 10。任务的输入数据大小为 0.5~2 Mbits, 所需的计算周期为 100~1 000 CPU cycles。边缘设备的覆盖范围为 200 m, 边缘设备之间的直线距离为 350 m。

首先, 对算法的收敛性进行评估。图 2 为本文提出的算法与传统粒子群算法对上述服务缓存和任务调度联合优化问题的求解情况对比。从实验结果可以看到, 在前 15 次迭代过程中收敛较快, 迭代 30 次后得到的效用函数值趋于平稳并找到最优解。这说明本文算法在全局优化能力方面具有优势, 相比传统的粒子群优化算法, 能够得到更小的效用函数值。

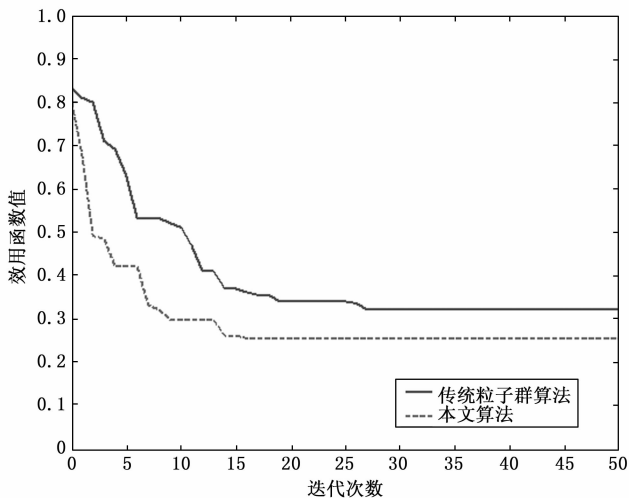


图 2 算法的收敛性对比

对比实验选取了远程云处理策略、随机策略、贪心算法、以及本文提出的算法在服务缓存和任务调度联合优化问题的应用效果进行了比较。本文算法中的参数初始化为: 种群规模初始化为 100, 最大迭代次数为 50。图 3 为边缘设备缓存空间利用率对比。所利用的服务器缓存空间越大, 移动用户任务的响应时间越短。由于远程云处理策略中所有移动用户的任务都是直接由边缘设备转发至云数据中心进行处理, 因此对应的 MEC 服务器不缓存任何任务, 会造成

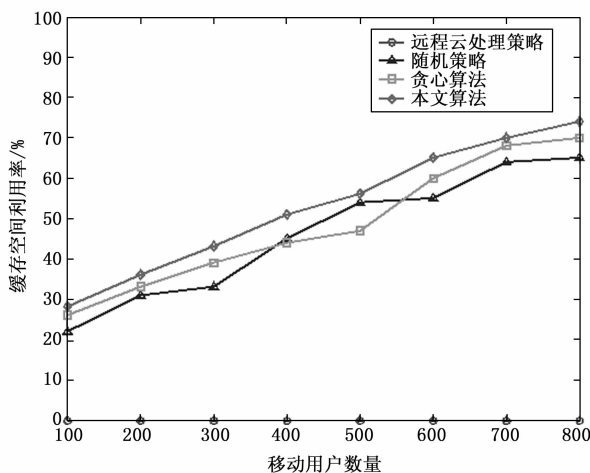


图 3 缓存空间利用率对比

成较大的传输延迟。随着移动用户数量的增加，任务数量越来越多，贪心算法和本文算法在缓存空间利用率上提升更快，这有利于实现卸载到边缘设备的任务能直接进行处理，减少任务的响应时间。

图 4 为边缘设备的负载均衡情况对比。从图 4 可以看出，在不同的用户规模条件下，本文算法和贪心算法能获得更好的负载均衡度。在远程云处理策略中，边缘设备负责将用户请求转发至远端云数据中心，其负载均衡度取决于覆盖范围内的用户数量。随机策略有时能获得较好的负载均衡度，但整体上不稳定。贪心算法在用户数量较大时负载均衡度偏大，说明其对边缘设备的利用不够均衡。

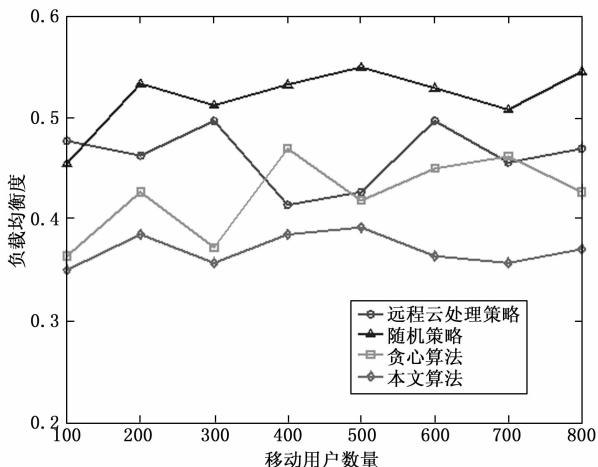


图 4 负载均衡情况对比

图 5 为执行时间的对比。可以看到使用随机策略，边缘设备的缓存任务和对移动用户的任务请求处理都是随机选择的，获得的目标值不稳定，且随机卸载策略的总延迟增长速度越快，相比贪心算法和本文算法在执行时间上的优化差距较大。在任务规模比较小的情况下，本文算法相比贪心算法的最优调度方案所需的总执行时间相差不明显。但是随着任务规模的增加，可以看到本文算法能够明显表现出在执行时间上的优势，当移动用户数在 400~800 时，

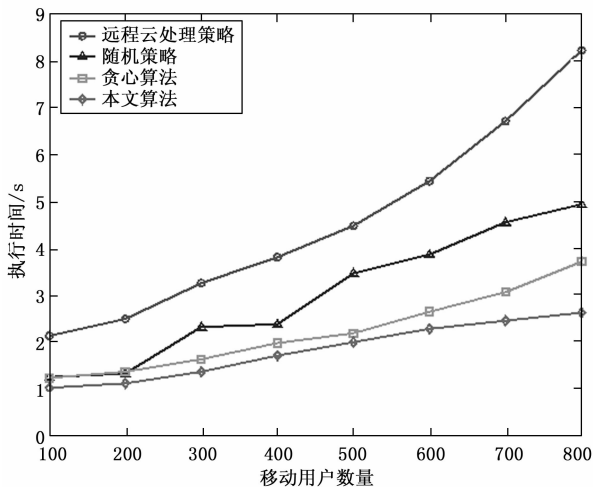


图 5 执行时间对比

本文提出的算法的总执行时间比贪心策略少 14.5%~32.1%。上述结果表明本文调度方案能够适应大规模任务调度，在较短的时间内完成用户任务。

4 结束语

针对任务卸载与服务缓存决策问题，本文建立数学模型并提出一种联合优化算法来求解任务执行时延约束条件下的服务缓存决策最优解。通过仿真实验，分析了边缘设备的缓存空间利用率、负载均衡、执行时间等指标下的效果。实验结果验证了本文提出的算法的有效性，与其他策略相比在提高卸载效率、减少任务完成时间和大规模任务调度的条件下都获得较好的性能。下一步工作，我们将继续探索和对服务缓存算法和卸载策略进行优化、完善。

参考文献:

- [1] HE B, MAO Y, GUO Z, et al. Comet: batched stream processing for data intensive distributed computing [C] // Acm Symposium on Cloud Computing. ACM, 2010.
- [2] LIU Y, PENG M, SHOU G, et al. Towards Edge Intelligence: Multi-Access Edge Computing for 5G and Internet of Things [J]. IEEE Internet of Things Journal, 2020, 7 (8): 6722-6747.
- [3] 施巍松, 孙辉, 曹杰, 等. 边缘计算: 万物互联时代新型计算模型 [J]. 计算机研究与发展, 2017, 54 (5): 907-924.
- [4] 吕华章, 陈丹, 范斌, 等. 边缘计算标准化进展与案例分析 [J]. 计算机研究与发展, 2018, 55 (3): 487-511.
- [5] 谢人超, 廉晓飞, 贾庆民, 等. 移动边缘计算卸载技术综述 [J]. 通信学报, 2018, 39 (11): 138-155.
- [6] TANG H, LI C, BAI J, et al. Dynamic resource allocation strategy for latency-critical and computation-intensive applications in cloud-edge environment [J]. Computer Communications, 2019, 134 (JAN.): 70-82.
- [7] LEE Z, YING W, WEN Z. A dynamic priority scheduling algorithm on service request scheduling in cloud computing [C] // International Conference on Electronic & Mechanical Engineering & Information Technology. IEEE, 2011.
- [8] LI C, BAI J, TANG J. Joint optimization of data placement and scheduling for improving user experience in edge computing [J]. Journal of Parallel and Distributed Computing, 2019, 125 (MAR.): 93-105.
- [9] 罗斌, 于波. 移动边缘计算中基于粒子群优化的计算卸载策略 [J]. 计算机应用, 2020, 40 (8): 2293-2298.
- [10] XU JIE, CHEN LIXING, ZHOU PAN. Joint service caching and task offloading for mobile edge computing in dense networks [C]. //Proc. of IEEE Conference on Computer Communications, 2018, pp. 207-215.
- [11] CHEN LIXING, SHEN CONG, ZHOU PAN, et al. Collaborative service placement for edge computing in dense small cell networks [J]. IEEE Transactions on Mobile Computing, 2020, 20 (2): 377-390.

(下转第 250 页)

原理, 结合非球面和像方远心技术, 分析设计成像系统; 最终设计出一款适用于 0.3 英寸的小尺寸、高亮度的投影系统。照明系统的数字微镜器件的照度均匀性达到 90.91%, 整个系统的光效率也达到了 88.45%; 成像系统长度为 53.9 mm, 焦距为 7.898 mm, 全视场达到 67°, 远心度优于 0.67°, 在 93 lp/mm 处调制传递函数大于 0.52, 垂轴色差在 0.5 pixels 以内, 畸变小于 1.0%。本文的研究工作取得了较好的结果, 但还有一定的优化空间, 后续将从节约成本、降低加工难度等方面入手。通过优化光学系统以缩减镜头材料种类, 减小批量生产成本; 再通过优化非球面系数, 进一步降低加工难度等。

参考文献:

- [1] 潘国涛, 闫钰锋, 于 信, 等. 矩形大口径激光光束质量评价光学系统设计 [J]. 中国光学, 2022, 15 (2): 306-317.
- [2] 刘 峰, 黄 镇, 陈 丽, 等. 基于 LED 的均匀照明投影光路设计 [J]. 广东工业大学学报, 2012 (3): 73-76.
- [3] 刘 旖, 林祖伦, 祁康成, 等. 基于 LED 光源的单片式 DLP 照明系统设计 [J]. 光电工程, 2011 (9): 99-103.
- [4] 芮大为. LED 微型投影光源的二次光学设计 [D]. 成都: 电子科技大学, 2012.
- [5] 陈 琛, 李维善, 张 禹, 等. 短焦数字投影镜头的光学设计 [J]. 光子学报, 2011, 40 (12): 1855-1859.
- [6] 刘宵婵, 陈 琛, 单 宏, 等. 基于 4k 分辨率小型投影机的鱼眼镜头设计 [J]. 应用光学, 2020, 41 (5): 1060-1066.
- [7] MWA B, Ak A, STB C, et al. Economic assessment for additive manufacturing of automotive end-use parts through digital light processing (DLP) [J]. CIRP Journal of Manufacturing Science and Technology, 2021, 35: 268-280.
- [8] 楼永坚, 黄晗骋, 林 斌. 基于数字光处理的结构光三维扫描系统的设计 [J]. 光学仪器, 2018, 40 (3): 65-70.
- [9] 罗隆恒, 林祖伦. 基于数字光处理的微型投影设备设计 [J].
- [12] POULARAKIS K, LLORCA J, TULION A M, et al. Joint service placement and request routing in multi-cell mobile edge computing networks [C]. //Proc. of IEEE Conference on Computer Communications, 2019, 10-18.
- [13] HE TING, KHAMFROUSH H, WANG SHIQIANG, et al. It's hard to share: Joint service placement and request scheduling in edge clouds with sharable and non-sharable resources [C]. //Proc. of 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS), 2018, 365-375.
- [14] TRAN T X, CHAN K, POMPILI D, COSTA. Cost-aware service caching and task offloading assignment in mobile-edge computing [C]. // Proc. of 2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), 2019, 1-9.
- [15] MA XIAO, ZHOU AO, ZHANG SHAN, et al. Cooperative service caching and workload scheduling in mobile edge computing [C]. //Proc. of IEEE Conference on Computer Communications, 2019, 2076-2085.

光学仪器, 2017, 39 (1): 50-55.

- [10] 于百华, 田志辉, 苏东奇, 等. 超短焦投影系统中自由曲面的多视场优化迭代设计方法 [J]. 光子学报, 2019, 48 (3): 46-53.
- [11] 王 康, 段 洁, 周见红, 等. 便携式投影仪投影物镜设计 [J]. 应用光学, 2019, 40 (2): 316-322.
- [12] YU BAIHUA, TIAN ZHIHUI, SU DONGQI, et al. Optical design of an ultra-short-focus projection system with low throw ratio based on a freeform surface mirror [J]. Chinese Optics, 2020, 13 (2): 363-371.
- [13] OHARA K, KUNZMAN A. Video processing technique for multi-media HDTV with digital micro-mirror array [J]. IEEE Transactions on Consumer Electronics, 1999, 45 (3): 604-610.
- [14] 肖 勇. 基于 DLP 的微型投影仪驱动电路设计与实现 [D]. 成都: 电子科技大学, 2014.
- [15] 张以漠. 应用光学 [M]. 北京: 电子工业出版社, 2008.
- [16] 屈碧香. 基于 RGB 三色 LED 光源照明的 DLP 投影系统光路设计 [D]. 杭州: 浙江大学, 2013.
- [17] 郁道银, 谈恒英. 工程光学 [M]. 北京: 机械工业出版社, 2011: 2-24.
- [18] 黄红林, 许 键. 基于 Zemax 的微型投影镜头设计 [J]. 光学仪器, 2016, 38 (1): 49-52.
- [19] 王显锋. 短焦微型投影系统的光学设计 [D]. 苏州: 苏州大学, 2018.
- [20] 李维善, 陈 琛, 张 禹, 等. 基于 ZEMAX 软件的 DLP 微型投影镜头的设计 [J]. 应用光学, 2011, 32 (6): 1121-1125.
- [21] 萧泽新. 工程光学设计: 第 2 版 [M]. 北京: 电子工业出版社, 2008.
- [22] 沈为民, 薛鸣球, 余建军. 大视场大相对孔径长波红外物镜 [J]. 光子学报, 2004 (4): 460-463.
- [23] 李 军, 蔡 屹, 谷 鹏, 等. 基于机器视觉的口岸车道闸机抬杆故障远程检测方法 [J]. 计算机测量与控制, 2022, 30 (3): 19-24.
- [16] KENNEDY J, EBERHART R. Particle Swarm Optimization [C]. //Proc. of IEEE International Conference on Neural Network. IEEE, 1995, 1942-1948.
- [17] 汪小威, 林 宁, 胡玉平. 移动边缘计算中利用 BPSO 的任务卸载策略 [J]. 计算机工程与设计, 2021, 42 (12): 3333-3341.
- [18] 周艳平, 蔡 素, 李金鹏. 一种粒子群和改进自适应差分进化混合算法及在生产调度中的应用 [J]. 计算机测量与控制, 2019, 27 (8): 227-230.
- [19] TANG Y, WANG Z, FANG J. Parameters identification of unknown delayed genetic regulatory network by a switching particle swarm optimization algorithm [J]. Expert Systems with Applications, 2011 (38): 2523-2535.
- [20] 张娟芝, 段中兴, 熊福力. 一种自适应粒子群算法在云资源调度中的应用 [J]. 计算机测量与控制, 2020, 28 (12): 217-221.
- [21] 张 钰, 王 蕾, 周红标, 等. 基于竞争学习的粒子群优化算法设计及应用 [J]. 计算机测量与控制, 2021, 29 (8): 182-189.