



小型微型计算机系统

Journal of Chinese Computer Systems

ISSN 1000-1220, CN 21-1106/TP

《小型微型计算机系统》网络首发论文

题目: 计算资源受限 MEC 中任务卸载与资源分配方法
作者: 鲜永菊, 宋青芸, 郭陈榕, 刘闯
收稿日期: 2021-01-18
网络首发日期: 2021-09-07
引用格式: 鲜永菊, 宋青芸, 郭陈榕, 刘闯. 计算资源受限 MEC 中任务卸载与资源分配方法. 小型微型计算机系统.
<https://kns.cnki.net/kcms/detail/21.1106.TP.20210906.1816.068.html>



网络首发: 在编辑部工作流程中, 稿件从录用到出版要经历录用定稿、排版定稿、整期汇编定稿等阶段。录用定稿指内容已经确定, 且通过同行评议、主编终审同意刊用的稿件。排版定稿指录用定稿按照期刊特定版式 (包括网络呈现版式) 排版后的稿件, 可暂不确定出版年、卷、期和页码。整期汇编定稿指出版年、卷、期、页码均已确定的印刷或数字出版的整期汇编稿件。录用定稿网络首发稿件内容必须符合《出版管理条例》和《期刊出版管理规定》的有关规定; 学术研究成果具有创新性、科学性和先进性, 符合编辑部对刊文的录用要求, 不存在学术不端行为及其他侵权行为; 稿件内容应基本符合国家有关书刊编辑、出版的技术标准, 正确使用和统一规范语言文字、符号、数字、外文字母、法定计量单位及地图标注等。为确保录用定稿网络首发的严肃性, 录用定稿一经发布, 不得修改论文题目、作者、机构名称和学术内容, 只可基于编辑规范进行少量文字的修改。

出版确认: 纸质期刊编辑部通过与《中国学术期刊 (光盘版)》电子杂志社有限公司签约, 在《中国学术期刊 (网络版)》出版传播平台上创办与纸质期刊内容一致的网络版, 以单篇或整期出版形式, 在印刷出版之前刊发论文的录用定稿、排版定稿、整期汇编定稿。因为《中国学术期刊 (网络版)》是国家新闻出版广电总局批准的网络连续型出版物 (ISSN 2096-4188, CN 11-6037/Z), 所以签约期刊的网络版上网络首发论文视为正式出版。

计算资源受限 MEC 中任务卸载与资源分配方法

鲜永菊, 宋青芸, 郭陈榕, 刘 闯

(重庆邮电大学 通信与信息工程学院, 重庆 400065)

E-mail: xianyj@cqupt.edu.cn

摘 要: 移动边缘计算(MEC)通过将计算中心下沉至网络边缘, 可以有效服务于任务计算。然而, MEC 拥有的计算资源并不是无限的, 这带来了诸多问题。文章针对计算资源有限 MEC 系统中的计算卸载, 分析了最大化 MEC 总收益的卸载与资源分配联合优化问题。首先通过 Stackelberg 模型来描述 MEC 与用户之间的交互, 使用差异化定价策略增加对卸载的约束, 然后将卸载问题转化为二元背包问题, 最后, 通过改进模拟退火算法分配计算资源, 并迭代得到最优方案。仿真结果表明, 所提方案可以最大程度提高用户卸载数量, 并有效地提高了 MEC 系统的收益。

关键词: 计算资源有限; 移动边缘计算; 卸载策略; 资源分配

中图分类号: TN929

文献标识码: A

Method of Task Offloading and Resource Allocation in MEC with Limited Computing Resources

XIAN Yong-ju, SONG Qing-yun, GUO Chen-rong, LIU Chuang

(School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

Abstract: Mobile edge computing (MEC) can effectively serve task computing by sinking the computing center to the edge of the network. However, the computing resources of MEC are not unlimited, which brings many problems. This paper analyses a joint optimization problem between unloading and resource allocation to maximize the total revenue of MEC for computational resources finite MEC systems. Firstly, the interaction between MEC and users is described through the Stackelberg model, the differential pricing strategy is used to increase the constraints on unloading, and then the unloading problem is transformed into a binary backpack problem. Finally, the computing resources are allocated by improving the simulated annealing algorithm, and the optimal scheme is obtained by iteration. Simulation results show that the proposed scheme can maximize the number of user unloading and effectively improve the benefits of MEC systems.

Key words: limited computing resources; mobile edge computing; unloading strategy; resource allocation

1 引 言

移动边缘计算(Mobile Edge Computing, MEC)技术通过将云数据中心下沉至靠近数据源端的接入网一侧, 能够近距离为用户提供计算、通信、存储等能力, 同时使得移动网络传输成本更低、效率更高、应用复杂度不受终端的限制^[1-3], 在 AR/VR、物联网(Internet of Things, IoT)、车联网等场景中有着广阔的应用前景。计算卸载技术作为 MEC 中十分重要的一环^[4], 为移动用户带来了海量计算能力, 拓展了用户处理任务的能力。

鉴于此, 众多学者针对计算卸载展开了研究, 其中卸载策略和资源分配是研究重点。Chouhan 与 Nguyen 等人采用部分卸载划分计算任务, 使任务同时在多端进行计算, 相较于二元卸载, 多端并行计算带来了效率的提升^[5-6]; Qin 等人通过博弈理论对单用户多 MEC 问题建模^[7], 为单个用户提供了多种卸载选择; 在 Qin 等人方案的基础上, Yi 等人提出

的卸载方案, 更加适合时延敏感场景^[8]; 另外, Zhang 等人通过结合机器学习完成了信道和计算资源的分配, 并分析了多种分类器学习效果上的差异^[9]。还有学者联合卸载策略与资源分配提出方案, 其中 Ren 等人考虑了随机任务到达模型, 提出 JCRM 方案, 以提高长期网络的稳定性^[10]; Hu 等人作为任务执行过程中的能耗问题提出了解决方案, 并设计基于方向交叉的遗传算法, 完成联合优化问题^[11]。由于资源分配与任务卸载往往是紧密联系的, 因此与单独的卸载策略或资源分配的方案相比, 联合方案往往能获得更佳的效果。

现有的工作大多都假设 MEC 计算资源是足够的, 请求卸载的任务都可以在满足约束条件下完成计算卸载, MEC 服务器不担心出现性能瓶颈。但随时间发展, 由基站硬件成本限制所导致 MEC 服务器计算资源有限的问题将日渐明显, 尤其在商圈, 机场等任务请求密集的网络中, MEC 服务器提供的计算资源往往不足以应对大量请求任务^[4]。现有的一些卸载方案, 在计算资源不足时会导致用户需求无法满足,

引发多用户竞争有限的 MEC 计算资源产生“死锁”问题^[12]，导致卸载用户数量下降；同时，在资源不足时，已有方案获得的系统效益往往较差。为改善这些问题，Ning 等人^[13-14]与 Wu 等人的方案均通过多 MEC 协作的方式扩展计算资源，将 MEC 无法完成的任务通过二次卸载的方式，交给其他仍有计算资源的 MEC 执行，但多 MEC 协作实际需满足的条件十分苛刻，实际的应用价值不大；Liu 等人提出一种卸载方案，在 MEC 没有额外计算资源可用时，通过价格制约卸载的方式保证最大化自身收益，并提出了两种定价策略，比较了方案之间的差异^[15]；Zhao 等人考虑任务主动缓存的方案^[16]，这种方案可以缓存热点计算任务，当同类任务请求时，可以快速响应且不占用 MEC 计算资源，这样可有效改善热点任务的计算，但随着任务种类的增多，缓存策略带来的收益将十分有限。

基于上述存在的问题，本文通过联合价格约束卸载与计算资源分配的方式，在计算资源受限 MEC 中实现总收益最大化。主要的工作有以下几个方面：

(1) 在计算资源受限 MEC 中通过调整价格制约任务卸载，并通过 Stackelberg 博弈建模于 MEC 与用户之间的交互。在博弈模型分析中，将最大化 MEC 系统收益的联合优化问题分解为多个子问题。

(2) 针对上述问题，为计算资源有限 MEC 提出了一种差异化定价方法，并在此基础上通过多任务卸载算法完成了任务卸载。

(3) 提出了一种改进模拟退火算法用于计算资源分配，算法通过控温策略保证广度搜索能力，通过资源重分配保证深度搜索能力。

2 系统模型

考虑在任务密集区域（如商圈，机场，学校等）的多用户单 MEC 网络，在一个卸载周期内，有大量用户通过无线链路请求计算卸载，区域内的 MEC 服务器负责接收请求的计算任务，完成计算后，将计算结果返回用户端。为简化问题，假设每个用户只请求一个计算任务，任务可以按位任意划分，使得任务可以在本地与 MEC 端并行运算。另外，假设卸载过程中每个用户拥有相同的信道带宽，且每个用户的信道占用不重叠的频率以同时将数据卸载到 MEC 服务器上，任务卸载时考虑准静态信道模型，信道在卸载周期内保持恒定不变，但可以在不同任务的卸载周期间发生变化，并假设卸载任务将在一定时间内完成计算。

以 $N = \{1, 2, \dots, N\}$ 表示 N 个用户的集合，用户 i 的任务通过二元组 $T_i(b_i, d_i)$ 表示，其中 b_i 表示任务的总数据量， d_i 表示计算 1bit 该任务数据所需要的 CPU 周期数；用户 i 卸载任务数据量的比例为 λ_i ， $\lambda_i \in [0, 1]$ ，即 $\lambda_i b_i$ 数据量的任务会被卸载到 MEC 服务器进行计算，使用集合 $V = \{\lambda_1, \lambda_2, \dots, \lambda_N\}$ 表示 N 个用户卸载比例集合。用户本地计

算能力为 $f_{loc,i}$ ，MEC 服务器为多任务提供有限的计算资源 f_{mec} ，使用集合 $W = \{f_{mec,1}, f_{mec,2}, \dots, f_{mec,N}\}$ 表示 MEC 为 N 个用户分配的计算资源集合。

为了确保卸载至 MEC 的任务可以在卸载周期内完成计算，现考虑一个实际的限制条件：设定 MEC 服务器在每个卸载周期内用于计算接收数据的总 CPU 周期上限为 F ^[15]，设定 F 可以确保在计算资源不足的情况下有序地处理到来的任务， F 值越大，对应的卸载周期越长。该约束条件为：

$$\sum_{i=1}^N \lambda_i d_i b_i \leq F \quad (1)$$

需要注意这里的 F 和 f_{mec} 分别是 MEC 服务器对于卸载 CPU 周期的总量与计算频率上限。

首先考虑用户与 MEC 之间的通信，任务的上传速率为： $r_i = B \log_2(1 + P_{tra,i} g_i / B N_o)$ ，其中 $P_{tra,i}$ 表示用户 i 的发送功率， g_i 表示用户 i 与 MEC 服务器之间的信道增益， N_o 表示信道单位噪声与干扰的功率谱密度。

其次，当任务卸载时，用户 i 本地执行部分任务花费的时间为： $t_{loc,i} = (1 - \lambda_i) b_i d_i / f_{loc,i}$ ；任务在 MEC 服务器端计算部分任务花费的时间为： $t_{mec,i} = t_{tra,i} + t_{exe,i}$ ，其中 $t_{tra,i} = \lambda_i b_i / r_i$ ，表示用户向 MEC 上传任务产生的时延， $t_{exe,i} = \lambda_i b_i d_i / f_{mec,i}$ 代表任务在 MEC 端的执行产生的时延；由于 MEC 向用户回传的计算结果数据量很小，因此回传时间忽略不计。最终用户 i 完成任务总时延为：

$$t_i = \max(t_{loc,i}, t_{mec,i}) \quad (2)$$

最后，用户 i 在本地执行部分任务消耗的能量为： $e_{loc,i} = \kappa_i b_i d_i (1 - \lambda_i) (f_{loc,i})^2$ ，其中 κ_i 是与移动用户的硬件架构相关的常数；用户 i 向 MEC 服务器发送任务需要的能耗为： $e_{tra,i} = P_{tra,i} \cdot t_{tra,i} = \lambda_i P_{tra,i} b_i / r_i$ 。最终用户 i 完成该任务的总能耗为：

$$e_i = e_{tra,i} + e_{loc,i} \quad (3)$$

3 优化问题分析及求解

3.1 Stackelberg 博弈分析

在 MEC 任务执行过程中，用户占用了 MEC 服务器的计算资源来处理任务，而 MEC 服务器必须确保其用于计算总卸载数据的可用 CPU 周期低于 F 。为了调整计算资源的需求和供给，MEC 服务器可对每个用户 i 的卸载数据的 CPU 周期 $b_i d_i$ 收费，以出售自身有限的资源。

对此，Stackelberg 博弈可用于建模 MEC 与用户之间的交互，Stackelberg 博弈模型是一种产量领导模型，可以反应非对称竞争关系，并将竞争者角色定为领导者与跟随者。本文将 MEC 服务器视为领导者，而用户视为跟随者。领导者先对跟随者的 CPU 周期制定价格；然后，跟随者将依据领导者制定的价格，独立计算其用于卸载的 CPU 周期，以分别进行本地计算和卸载。

将 MEC 给用户 i 单位 CPU 周期的价格定义为 μ_i ， $U = \{\mu_1, \mu_2, \dots, \mu_N\}$ 表示所有用户的单价集合，MEC 最大化收益问题表示为（主问题）：

$$\begin{aligned} \mathbf{P1}: \max Q(U, V, W) &= \sum_{i=1}^N \mu_i \lambda_i d_i b_i \\ \text{s.t. C1: } \mu_i &> 0 \\ \text{C2: } \sum_{i=1}^N f_{mec,i} &\leq f_{mec} \\ \text{C3: } \sum_{i=1}^N \lambda_i d_i b_i &\leq F \end{aligned} \quad (4)$$

式(4)中约束条件 C1 确保 MEC 服务器提供的价格为正数，C2 确保 MEC 在并行运算时为每个用户分配的频率不大于总频率，C3 确保 MEC 接收 CPU 总周期小于上限值。对于用户而言，其主要目的是降低完成任务所需要付出的成本。成本由完成任务付出的时延、能耗以及费用的加权和表示^[15]。这个问题表示为（从问题）：

$$\begin{aligned} \mathbf{P2}: \min U_i(\lambda_i) &= \alpha_i t_i + \beta_i e_i + \gamma_i \mu_i \lambda_i d_i b_i \\ \text{s.t. C1: } 0 &\leq \lambda_i \leq 1 \\ \text{C2: } \alpha_i + \beta_i + \gamma_i &= 1 \end{aligned} \quad (5)$$

其中 α_i 、 β_i 和 γ_i 分别表示用户 i 对于执行任务产生的时延、能耗以及费用成本的权重，不同用户权重不同。Stackelberg 博弈中的问题 **P1** 和 **P2** 以一种复杂的方式耦合在一起，即 MEC 的定价策略与资源分配会影响用户的卸载，而用户卸载又会反过来影响边缘云的收益。

考虑问题中 MEC 与用户之间的博弈，每个用户都可通过给定价格 μ_i 以及 $f_{mec,i}$ 求解问题 **P2**，独立决定其卸载比例。知道了每个用户的卸载比例，MEC 通过求解问题 **P1** 来设置其最佳单价集合 U 、卸载比例集合 V 以及资源分配集合 W ，以上过程称为逆向归纳。由于问题是多目标优化，为降低求解难度，将原问题分为三个子问题，分别求解定价策略、卸载策略以及资源分配问题。

3.2 差异化定价策略

本节分析 MEC 的定价策略，假设分配给用户 i 的 $f_{mec,i}$ 已知。先分析跟随者的行为，根据式(5)，用户 i 的总成本可写为 λ_i 的分段函数：

$$U_i = \begin{cases} (\gamma_i \mu_i d_i + \frac{\beta_i P_{tra,i}}{r_i} - \beta_i \kappa_i d_i f_{loc,i}^2 - \frac{\alpha_i d_i}{f_{loc,i}}) \lambda_i b_i + \frac{\alpha_i d_i b_i}{f_{loc,i}} + \beta_i \kappa_i b_i d_i f_{loc,i}^2 & 0 \leq \lambda_i \leq \varphi_i \\ (\gamma_i \mu_i d_i + \frac{\alpha_i}{r_i} + \frac{\alpha_i d_i}{f_{mec,i}} + \frac{\beta_i P_{tra,i}}{r_i} - \beta_i \kappa_i d_i f_{loc,i}^2) \lambda_i b_i + \beta_i \kappa_i b_i d_i f_{loc,i}^2 & \varphi_i < \lambda_i \leq 1 \end{cases} \quad (6)$$

其中 $\varphi_i = 1 / (1 + f_{loc,i} / d_i r_i + f_{loc,i} / f_{mec})$ ，是在区间 $[0,1]$ 内的一个卸载比例。公式(6)的前半段表示用户在本地计算的时延较长，后半段表示卸载至 MEC 服务器的计算时延较长。

当 $\lambda_i = \varphi_i$ 时表示本地与卸载计算时延相等。为了求解 **P2**，对 U_i 一阶求导，可得卸载比例与单价的关系，由(7)式给出：

$$\lambda_i = \begin{cases} 1 & 0 \leq \mu_i \leq G_{1,i} \\ \varphi_i & G_{1,i} < \mu_i \leq G_{2,i} \\ 0 & G_{2,i} < \mu_i \end{cases} \quad (7)$$

此时得到的 λ_i 为用户 i 的最佳卸载比例，用户将根据该比例卸载任务，其中：

$$G_{1,i} = -\frac{\beta_i P_{tra,i}}{\gamma_i r_i d_i} + \frac{\beta_i \kappa_i f_{loc,i}^2}{\gamma_i} - \frac{\alpha_i}{\gamma_i r_i d_i} - \frac{\alpha_i}{\gamma_i f_{mec,i}} \quad (8)$$

$$G_{2,i} = -\frac{\beta_i P_{tra,i}}{\gamma_i r_i d_i} + \frac{\beta_i \kappa_i f_{loc,i}^2}{\gamma_i} + \frac{\alpha_i}{\gamma_i f_{loc,i}} \quad (9)$$

此外，式(7)将作为博弈中的反应函数，被领导者观察到。

接着分析 MEC 服务器行为，领导者根据观察到的反应函数调整 μ_i ，MEC 的定价策略由定理一给出：

定理一：为使收益最大化，计算资源受限 MEC 需将 μ_i 定为 $G_{2,i}$ 或正无穷。

证明：为了最大化 MEC 系统的利润，MEC 服务器应在保证用户卸载量不变的情况下，尽可能提高售价，因此 MEC 服务器应将单价定在反应函数每个分段右界。当满足式(7)第一段时， μ_i 应为 $G_{1,i}$ ，此时单价较小，用户全部卸载；同理，当满足第二段时， μ_i 应为 $G_{2,i}$ ，此时单价较大，用户的卸载比例为 φ_i ；当定价位于第三段时， μ_i 应为正无穷，此时用户不卸载。显然，定价为 $G_{1,i}$ 不适合计算资源有限系统；同时，在信道资源充足时，用于传输任务的能耗将小于本地计算该任务带来的能耗^[11]，因此 $G_{2,i}$ 为正数，保证了定价的合理性。故计算资源受限 MEC 应将价格定在 $G_{2,i}$ 或无穷高。

证毕

3.3 多用户卸载策略

定价策略决定后，MEC 需限制多用户卸载的总 CPU 周期小于 F ，通过引入二进制决策变量集合 x 用来表示用户的卸载决策：

$$\lambda_i = \varphi_i x_i \quad (10)$$

其中二进制数 x_i 定义为：

$$x_i = \begin{cases} 0 & G_{2,i} < \mu_i \\ 1 & 0 \leq \mu_i \leq G_{2,i} \end{cases} \quad (11)$$

此时问题 **P1** 被重写为：

$$\begin{aligned} \mathbf{P3}: \max Q(x, U) &= \sum_{i=1}^N \mu_i \varphi_i x_i b_i d_i \\ \text{s.t. C1: } \sum_{i=1}^N \varphi_i x_i d_i b_i &\leq F \end{aligned} \quad (12)$$

根据定理一，当 $x_i = 1$ 时，最优价格由 $G_{2,i}$ 给出。最优价格与用户任务的信息相关，这些信息可以在当用户向 MEC 服务器发起卸载请求时获取。问题 **P3** 可转化为：

$$\begin{aligned} \mathbf{P3'}: \max Q(\mathbf{x}) &= \sum_{i=1}^N G_{2,i} \varphi_i b_i d_i \\ \text{s.t. C1: } &\sum_{i=1}^N \varphi_i x_i d_i b_i \leq F \end{aligned} \quad (13)$$

问题 **P3'** 可等效为重量为 $\varphi_i b_i d_i$ ，价值为 $G_{2,i} \varphi_i b_i d_i$ ，重量限制为 F 的非整数二元背包问题，该问题是 NP-complete 的。对问题的求解分为两步，首先设置一个较大的步长 F_d ，将重量与 F 进行近似处理，将非整数 0-1 背包问题转化为整数 0-1 背包问题，同时减小了问题的规模；接着，通过动态规划思想对问题求解，表 1 为多用户卸载决策算法实现步骤。

表 1 多用户卸载决策算法

Table.1 Multi-user offloading decision algorithm

输入: 用户集 N ，定价策略集 G_2 ，CPU 周期上限为 F ，步长 F_d
 初始化: $L = F / F_d$;
 for $i = 1 : N$
 for $j = 1 : L$
 if ($j < \varphi_i b_i d_i / F_d$)
 $V(i, j) = V(i-1, j)$;
 else
 if ($V(i-1, j) > V(i-1, j - \varphi_i b_i d_i / F_d) + G_{2,i} \varphi_i b_i d_i$)
 $V(i, j) = V(i-1, j)$;
 else
 $V(i, j) = V(i-1, j - \varphi_i b_i d_i / F_d) + G_{2,i} \varphi_i b_i d_i$;
 end if
 end if
 end for
 end for
 最大收益 $Q^* = V(N, L)$
输出: 最大收益 Q^* ，卸载决策变量 \mathbf{x}

3.4 基于模拟退火思想的计算资源分配

3.2 节与 3.3 节中的定价策略以及卸载策略均是在计算资源分配已告知用户的条件下得出的，接下来分析计算资源分配问题。

计算资源分配的难点在于多任务决策变量与资源分配变量是耦合在一起的。一方面决策变量需要计算资源分配的结果；另一方面，MEC 实际只需为决策变量 \mathbf{x} 中值为 1 的用户分配计算资源，资源分配同样依赖决策变量的结果。由于以上原因，本文通过模拟退火算法随机寻优的方式完成计算资源分配。模拟退火算法是依靠随机数方法迭代求解的寻优算法，伴随着算法中温度不断下降，解状态会逐渐稳定，这个过程中有概率发生突跳，使得解状态重新进入不稳定状态。

在最大化收益问题中，由于部分用户不卸载，MEC 服务器不需要为这些用户分配计算资源，其计算资源分配应为 0。含 0 计算资源分配解极易导致模拟退火算法陷入局部最优解^[17]，并很难在有限的迭代中跳出局部最优解。为解决该问题，对模拟退火算法做出两点改进：

(1) 计算资源集合 \mathbf{W} 作为算法的解状态，算法中决策变量确定后，需对决策变量中值为 1 的用户进行计算资源重分配，通过求解凸优化问题解决。

(2) 算法加入控温策略，并且算法只考虑基于温度的单层循环，当温度下降至小于 1，算法停止。

改进(1)可确保解的精度，假设根据卸载策略确定卸载的用户集合 $\mathbf{M} = \{1, 2, \dots, M\}$ ， $j \in \mathbf{M}$ 代表其中第 j 个用户， $\mathbf{W}' = \{f_{mec,1}, f_{mec,2}, \dots, f_{mec,M}\}$ 表示 M 个用户的计算资源的集合。此时用户 j 将以卸载比例 φ_j 卸载任务，问题 **P1** 可重写为：

$$\begin{aligned} \mathbf{P4}: \max Q(\mathbf{W}') &= \sum_{j=1}^M \mu_j \frac{1}{1 + \frac{f_{loc,j}}{d_j r_j} + \frac{f_{loc,j}}{f_{mec,j}}} b_j d_j \\ \text{s.t. C1: } &\mu_j > 0 \\ \text{C2: } &\sum_{j=1}^M f_{mec,j} = f_{mec} \\ \text{C3: } &\sum_{j=1}^M \frac{1}{1 + \frac{f_{loc,j}}{d_j r_j} + \frac{f_{loc,j}}{f_{mec,j}}} b_j d_j \leq F \end{aligned} \quad (14)$$

优化问题 **P4** 是分数规划问题，对其进行变量替换，令 $Y_j = f_{mec,j} / [(1 + f_{loc,j} / r_j b_j) f_{mec,j} + f_{loc,j}]$ ，同时令 $Z_j = 1 / [(1 + f_{loc,j} / r_j b_j) f_{mec,j} + f_{loc,j}]$ ，得到集合 $\mathbf{Y} = \{Y_1, Y_2, \dots, Y_M\}$ 。将原问题 **P4** 转化为：

$$\begin{aligned} \mathbf{P4'}: \max Q(\mathbf{Y}) &= \sum_{j=1}^M \mu_j b_j d_j Y_j \\ \text{s.t. C1: } &\mu_j > 0 \\ \text{C2: } &\sum_{j=1}^M Y_j = Z_j f_{mec} \\ \text{C3: } &\sum_{j=1}^M Y_j b_j d_j \leq F \\ \text{C4: } &Z_j \geq 0 \\ \text{C5: } &(1 + \frac{f_{loc,j}}{r_j d_j}) Y_j + f_{loc,j} Z_j = 1 \end{aligned} \quad (15)$$

问题 **P4'** 的函数式与约束条件均为凸函数，所以问题 **P4'** 是一个凸优化问题，考虑到约束条件里有不等式约束，可通过内点法对其求解^[18]。

改进(2)可降低算法搜索次数，同时增加算法解空间的广度搜索能力。只考虑基于温度的单层循环，将大幅减少迭代次数，但这并没有丢失模拟退火算法的核心思想。

此外，根据 Metropolis 准则，当前解向一次较差解的移动的概率随着温度的变化而变化，根据这一特征，算法从较大的温度开始，采用相对平缓的温度下降公式 $T_k = T_{k-1} \times \eta$ ，（ η 为降温系数）保证了算法开始有较大的概率接受差解。控温策略是当解没有更新时，以 $T_k / 2T_0 > \text{random}[0,1]$ 的概率保持温度不变，保持算法接受较差解的概率，提高算法的广度搜索能力。表 2 为改进模拟退火算法的具体步骤。

表 2 改进模拟退火算法

Table.2 Improved simulated annealing algorithm

输入: 用户信息，初始资源分配集合 \mathbf{W}_0 ，初始温度 T_0
 初始化: $k \leftarrow 0$ ， $\sigma_0 \leftarrow \mathbf{0}$;
 repeat

```

 $k \leftarrow k+1$ ;
While (true)
    随机产生置乱集合  $\sigma_k$ ;
     $\mathbf{W}_k \leftarrow \mathbf{W}_{k-1} + \sigma_k$ ;
    if ( $\mathbf{W}_k$  内元素满足约束 C2)
        break;
    end if
end while
根据定理一得到单价集合  $\mathbf{U}_k$ ，多用户卸载决策算法得到  $\mathbf{x}_k$ ;
对  $\mathbf{x}_k$  中值为 1 的用户通过求解凸优化得到计算资源集  $\mathbf{W}'_k$ ;
计算目标函数  $q$ ;
根据 Metropolis 准则与控温策略调整解状态  $\mathbf{W}_k$ ，温度  $T_k$ 。
if ( $q > q$ )
     $\mathbf{W}_{best} \leftarrow \mathbf{W}'_k$ ， $q \leftarrow q$ ;
end if
until  $T \leq 1$ 
输出：资源分配  $\mathbf{W}_{best}$ ，总收益  $q$ 

```

伴随着温度下降，算法在每次迭代中，首先通过随机置乱产生一组资源分配变量；接着根据公式(9)以及表 1 多用户卸载决策算法得到卸载决策变量；最后通过求解(15)式的凸优化问题完成计算资源再分配。直至温度下降小于 1，迭代结束，否则更新资源分配变量进入下一次迭代。

根据控温策略与温度下降公式，温度可在常数时间内下降至 1，即迭代次数 k 为常数；多用户卸载决策算法需要遍历整个二维数组，时间复杂度为 $O(NL)$ ；解向量更新的时间复杂度为 $O(N)$ 。故改进模拟退火算法总的时间复杂度为 $O(kNL + kN)$ 。

4 仿真分析

本文仿真分析基于 MATLAB 平台，参考文献[15]的仿真参数设置：信道总带宽 $B=1\text{MHz}$ ，信道噪声与干扰的功率谱密度为 -174dBm/Hz ，信道增益在 $[-50, -30]\text{dBm}$ 内均匀分布。用户 i 本地的 CPU 频率 $f_{loc,i}$ 在 $[0.1, 1]\text{GHz}$ 中随机取值；任务的数据大小 b_i 在 $[100, 500]\text{KB}$ 中随机取值；单位任务计算量 d_i 在 $[500, 1500]\text{cycles/bit}$ 中随机取值。默认状态下，算法的初始温度为 40，用户数量 $N=40$ ，用户 i 的发射功率 $P_{tra,i}=0.1\text{W}$ ，系数 $\kappa=10^{-27}$ ，MEC 服务器提供的 CPU 频率 $f_{mec}=100\text{GHz}$ ，CPU 周期上限为 $F=6 \times 10^9 \text{ cycles/slot}$ ，降温系数 $\eta=0.97$ 。

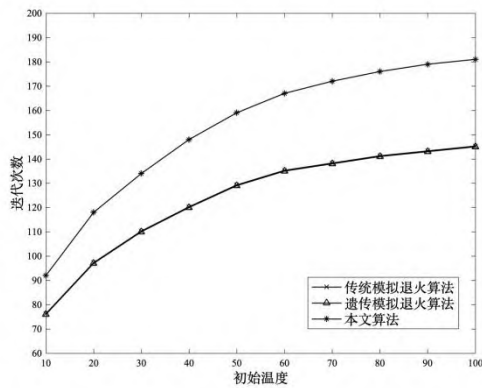


图 1 初始温度对迭代次数的影响

Fig.1 Relationship between initial temperature and number of

iterations

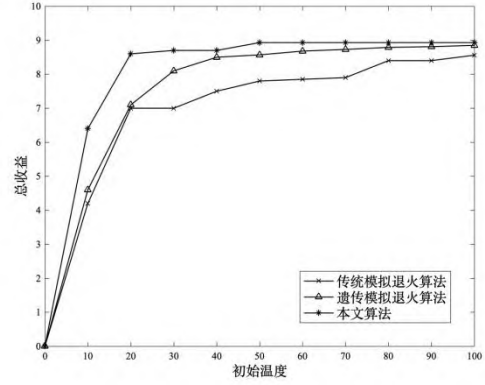


图 2 初始温度对总收益的影响

Fig.2 User requests and the actual number of users to uninstall relations

首先验证本文对模拟退火算法的改进对问题解的影响，图 1 图 2 分别是受初始温度的影响，算法迭代次数与问题解的变化情况。仿真对比了与传统模拟退火算法以及文献[19]中遗传模拟退火算法。由于本文算法增加了控温策略，整体迭代次数多于传统模拟退火算法，但在可接受的范围内。关于算法解质量的对比，本文算法可在较低的初始温度达到解的稳定，并且解的质量较优；传统算法容易陷入局部最优解，解的质量较差；遗传退火算法在模拟退火算法中加入适应性替换方法，可以改善算法过早收敛的问题，但对本文问题的求解效果并不是很理想。因此，改进算法在处理本文联合优化问题有较好的效果。

资源有限 MEC 中所面临的问题主要是由于“死锁”导致卸载用户数量降低，以及解的质量不佳两个问题，接下来分别验证两点，并与基于贪心的卸载算法以及文献[15]中提出的统一卸载算法、非统一卸载算法进行对比。

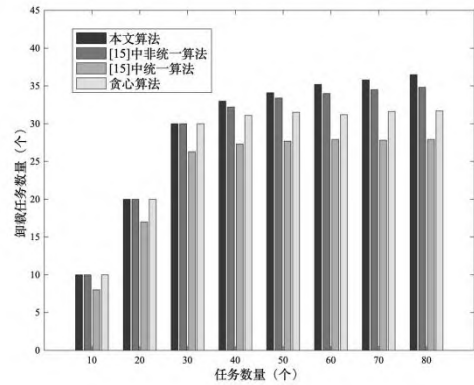


图 3 请求用户数对实际卸载用户数的影响

Fig.3 User requests and the actual number of users to uninstall relations

图 3 表示总卸载用户数与请求卸载用户数量的关系。当请求卸载用户数量较少的时候，多数算法都能保证所有用户

完成卸载。随着请求卸载的用户数量上升，总卸载用户数也呈上升趋势。由于受到 F 的限制，实际卸载用户数量到达 30 左右时，将无法容纳更多的用户完成卸载。本文算法联合卸载与资源分配，有利于调整多用户的卸载行为，避免了多用户竞争资源所引发的问题，提高了实际卸载的用户数量，而贪心算法与非统一算法不同程度受到“死锁”问题的影响，用户卸载数量较低，统一算法通过价格限制了很多用户的卸载，因此卸载用户数量最低。

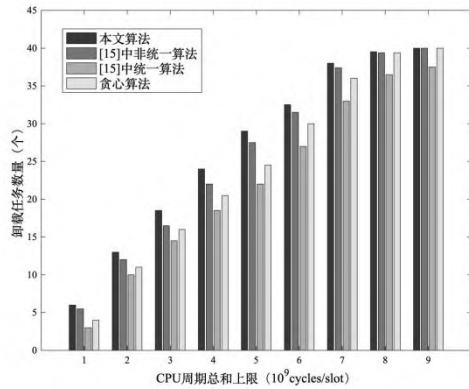


图 4 CPU 周期上限对实际卸载用户数的影响

Fig.4 Effect of the upper limit of CPU cycles on the actual number of users uninstalled

图 4 表示 F 值变化对 MEC 实际卸载用户数量的影响，随着 F 的增加，实际卸载的用户数量也随之增加。当 F 值较低时，本文算法相比其他算法卸载用户数提升较明显，这是因为本文算法结合了计算资源分配，计算资源影响用户的卸载量，使得更多用户卸载执行。随着 F 值增加，四种算法基本都可以满足所有用户卸载。

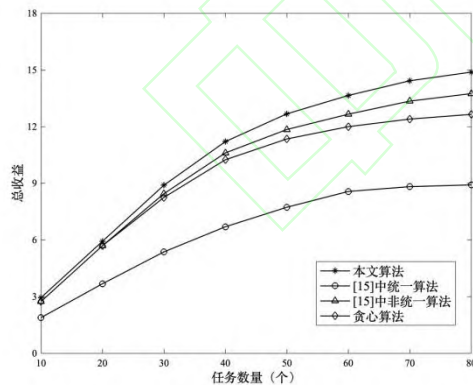


图 5 用户数量对总收益的影响

Fig.5 The number of users relationship with total revenue

图 5 表示请求用户数量与 MEC 系统收益的关系。仿真结果表示随着请求卸载的用户数增加，四种算法得到的总收益呈上升趋势。当用户数量小于 30，收益上升的趋势近似线性增长，这是因为用户数量较小时，MEC 服务器所提供的计算资源可以满足所有的用户都进行卸载。当用户数量增多时，四种策略的总收益上升趋势趋于平缓，这是由于用户

任务的计算量达到了 F 上限，限制了一些用户任务的卸载，本文算法中联合了定价、卸载与计算资源分配，优化了用户卸载比例，可以得到最佳用户卸载组合，获得最大的收益。

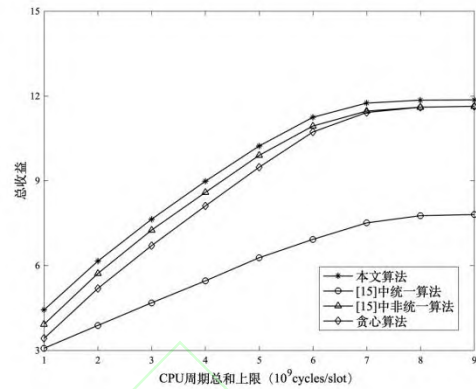


图 6 CPU 周期上限对总收益的影响

Fig.6 Relationship between the upper limit of CPU cycles and total revenue

图 6 表示 F 值变化对 MEC 系统总收益的影响。随着 F 的增加，可以卸载到边缘云的任务数量也随之增加，四种算法得到的总收益都呈现整体上升的趋势。当 F 值较低时，本文算法相较其他 3 种算法的提升比较明显，随着 F 值的增加，本文算法、贪心算法与非统一算法都获得较好的收益，但本文算法为多用户提供了更合理的计算资源，有最高的总收益。另外统一算法只考虑单一定价方案，导致任务卸载门限变高，因此总收益最低。

5 结论

本文分析了计算资源有限 MEC 中任务卸载和资源分配问题。通过价格约束的方法限制部分用户的卸载，同时为 MEC 设定 CPU 周期接收上限以保证任务可持续执行，在此基础上解决了最大化 MEC 系统总收益的联合优化问题：首先通过 Stackelberg 博弈来模拟 MEC 与用户之间的交互，通过改进模拟退火算法完成计算资源分配，并在每次迭代中对用户制定差异化定价、卸载决策以及计算资源重分配。最终，算法通过迭代，高效地得到了最佳卸载与资源分配方案，最大程度提高了卸载用户数量，同时有效提高了 MEC 系统的总收益。

References:

- [1] Satyanarayanan M, Bahl P, Caceres R, et al. The case for VM-based cloudlets in mobile computing[J]. IEEE Pervasive Computing, 2009, 8(4): 14-23.
- [2] Baktir A C, Ozgovde A, Ersoy C. How can edge computing benefit from software-defined networking: a survey, use cases & future directions[J]. IEEE

- Communications Surveys & Tutorials, 2017, 15(4): 2359-2391.
- [3] Tran T X, Hajisami A, Pandey P, et al. Collaborative mobile edge computing in 5G networks: new paradigms, scenarios, and challenges[J]. IEEE Communications Magazine, 2017, 55(4):54-61.
- [4] Sun Xiang, Ansari N. EdgeIoT: mobile edge computing for the internet of things[J]. IEEE Communications Magazine, 2016, 54(12):22-29.
- [5] Chouhan S. Energy optimal partial computation offloading framework for mobile devices in multi-access edge computing[C]//International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Split, Croatia: IEEE Press, 2019: 1-6.
- [6] Nguyen P, Ha V N, Le L B. Computation offloading and resource allocation for backhaul limited cooperative MEC systems[C]//90th Vehicular Technology Conference (VTC2019-Fall), Honolulu, HI, USA: IEEE Press, 2019: 1-6.
- [7] Qin An, Cai Cheng-cheng, Wang Qin, et al. Game theoretical multi-user computation offloading for mobile-edge cloud computing[C]//Conference on Multimedia Information Processing and Retrieval (MIPR), San Jose, CA, USA: IEEE Press, 2019: 328-332.
- [8] Yi Chang-yan, Cai Jun, Su Zhou. A multi-user mobile computation offloading and transmission scheduling mechanism for delay-sensitive applications[J]. IEEE Transactions on Mobile Computing, 2020, 19(1): 29-43.
- [9] Zhang Yong, Zhou Xiao-min, Teng Ying-lei, et al. Resource allocation for multi-user MEC system: machine learning approaches[C]//International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA: IEEE Press, 2018: 794-799.
- [10] Ren J, Mahfujul M K, Lyu F, et al. Joint channel allocation and resource management for stochastic computation offloading in MEC[J]. IEEE Transactions on Vehicular Technology, 2020, 69(8): 1528-1536.
- [11] Hu Shi-hong, Li Guang-hui. Dynamic request scheduling optimization in mobile edge computing for IoT applications[J]. IEEE Internet of Things Journal, 2020, 7(2): 1426-1437.
- [12] Ugwuanyi E E, Ghosh S, Iqbal M, et al. Reliable resource provisioning using bankers' deadlock avoidance algorithm in MEC for industrial IoT[J] IEEE Access, 2018, 6: 43327-43335.
- [13] Ning Zhao-long, Dong Pei-ran, Kong Xiang-jie, et al. A cooperative partial computation offloading scheme for mobile edge computing enabled internet of things[J]. IEEE Internet of Things Journal, 2019, 6(3): 4804-4814.
- [14] Wu Jin-ze, Cao Zhi-ying, Zhang Ying-jun, et al. Edge-cloud collaborative computation offloading model based on improved partial swarm optimization in MEC[C]//25th International Conference on Parallel and Distributed Systems (ICPADS) Tianjin, China: IEEE Press, 2019: 959-962.
- [15] Liu Meng-yu, Liu Yuan. Price-based distributed offloading for mobile-edge computing with computation capacity constraints[J]. IEEE Wireless Communications Letters, 2018, 7(3): 420-423.
- [16] Zhao Hong-yu, Wang Ying, Sun Rui-jing. Task proactive caching based computation offloading and resource allocation in mobile-edge computing systems[C]//14th International Wireless Communications & Mobile Computing Conference (IWCMC), Limassol: IEEE Press, 2018: 232-237.
- [17] Lu Yu-ting, Lin Yu-you, Peng Qiao-zi, et al. Overview of simulated annealing algorithm improvement and parameter exploration[J]. College Mathematics, 2015, 31(6): 96-103.
- [18] Liu Peng, Li Jian-dong, Li Hong-yan, et al. Convex optimisation-based joint channel and power allocation scheme for orthogonal frequency division multiple access networks[J]. IET Communications, 2015, 9(1): 28-32.
- [19] Chen Zhuo, Feng Gang, Liu Yi-jing, et al. Virtual network function deployment strategy based on improved genetic simulated annealing algorithm in MEC[J]. Journal of Communications, 2020, 39(4): 74-84.

附中文参考文献:

- [17] 卢宇婷, 林禹攸, 彭乔姿, 等. 模拟退火算法改进综述及参数探究[J]. 大学数学, 2015, 31(06):96-103.
- [19] 陈卓, 冯钢, 刘怡静, 等. MEC 中基于改进遗传模拟退火算法的虚拟网络功能部署策略[J]. 通信学报, 2020, 39(4):74-84.