

移动边缘计算中计算卸载与资源分配的联合优化策略^①

龙 隆^{②***} 刘子辰^{*} 石晶林^{***} 周一青^{③***} 邱大伟^{***} 徐顺清^{***}

(^{*}中国科学院计算技术研究所移动计算与新型终端北京市重点实验室 北京 100090)

(^{**}中国科学院大学 北京 100049)

摘要 针对移动边缘计算(MEC)网络中计算能力不足的移动终端在处理低时延、高可靠应用而产生的高时延问题,提出一种基于博弈论的计算卸载与资源分配的联合优化策略。该策略首先将所有移动终端的卸载决策与资源优化问题构建成非合作博弈模型并证明其纳什均衡的存在性;其次给定移动终端的卸载决策,采用拉格朗日乘子法获取最优的上行、下行频谱资源以及计算资源;最后采用迭代法获取纳什均衡解。仿真结果表明,本文所提算法能够在资源有限的条件下最小化移动终端的任务时延,提升终端用户的服务体验。

关键词 移动边缘计算(MEC); 博弈论; 资源分配; 计算卸载

0 引言

随着移动通信的高速发展以及移动终端的快速普及,许多新型应用随之产生,如虚拟现实、增强现实以及自动驾驶等^[1-5]。而这类需要具有低时延高可靠特点的应用对移动终端的计算能力提出较高要求。由于计算能力有限的移动终端在处理这类应用时将产生较高的时延进而影响终端用户的服务体验,因此如何降低应用处理时延、提升终端用户的服务体验是目前亟需解决的关键问题之一^[6]。针对以上问题,移动云计算(mobile cloud computing, MCC)技术被提出。MCC 的目标是将云端丰富的计算资源扩展至资源受限的移动终端,从而增强移动终端潜在的计算能力^[7-10]。为实现该目标,移动终端需要将计算密集的任务通过无线接入的方式迁移至云服务器。尽管这种方式可以降低移动终端的负载但也存在明显的缺陷,即移动终端与云服务器较远的距离以及大量的终端业务请求将导致网络时延的增加,从而降低终端用户服务体验。

为了进一步降低网络时延,移动边缘计算(mobile edge computing, MEC)作为一种新的技术被提出。在 MEC 系统中,由基站与边缘服务器组成的服务节点就近为移动终端提供计算、通信与存储服务^[11,12]。然而与云服务器相比,服务节点的计算资源有限,过多的计算任务将为服务节点带来额外的负载从而影响网络时延。因此为解决上述问题,需要设计一种新的 MEC 计算卸载与资源分配的联合优化策略。

目前,基于 MEC 的卸载策略与资源分配方面已有相关研究。文献[13]以最小化移动终端的能耗与时延为目标,对移动终端的卸载策略、计算资源以及传输功率进行联合优化。该研究优化了移动终端的计算能力、功耗以及卸载策略,但并未考虑云端计算资源的优化。文献[13]是在 MCC 应用场景下对时延与能耗进行研究,然而在该场景下移动终端与 MCC 较远的物理距离可能会导致额外的时延,进而影响用户的服务体验。文献[14]则提出了在 MEC 网络下一种基于排队论的低复杂度算法,该算法通过优化频谱资源与边缘服务器的计算资源以最小化

① 国家自然科学基金(61571425),北京市自然科学基金(L172049)和国家科技重大专项(2018ZX03001017)资助项目。

② 男,1988 年生,博士生;研究方向:无线通信,边缘计算;E-mail: longlong@ict.ac.cn

③ 通信作者,E-mail: zhouyiqing@ict.ac.cn

(收稿日期:2019-08-31)

移动终端的能耗与时延。文献[15]则提出了一种上行链路等功率分配方案,基于该方案对设备能耗进行优化。此外,由于博弈论是一种有效的分布式方法,因此广泛应用于解决 MEC 场景下的资源优化分配问题^[16-18]。文献[19]则提出了基于博弈论的计算资源与频谱资源联合优化算法,解决了多终端用户的计算卸载与资源分配问题。尽管文献[14]与文献[19]采用不同的方法降低了终端的时延与能耗,但两者都忽略了下行频谱资源对终端时延的影响,而这在实际应用场景中并不总是成立的。例如在增强现实游戏应用场景中,移动用户将实时发送视频数据,而服务端将对收到的视频数据进行识别并经过渲染将视频返回至用户端。在此过程中,移动用户的任务时延不但受到数据上传、计算时延的影响,同样受到下行数据传输时延的影响,因此对于这类应用场景,下行链路资源的优化显得十分重要。

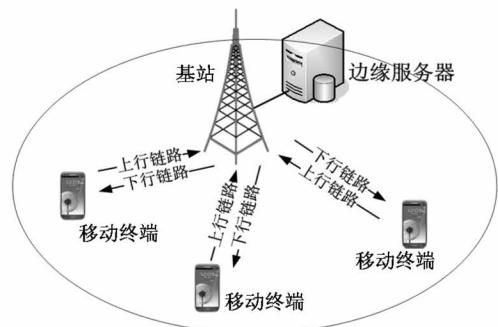
本文在 MEC 系统下,以最小化所有移动终端的任务时延为目标,提出一种基于博弈论的计算卸载与资源分配的联合优化算法(computation offloading and resource allocation game, CORAG)。该算法首先将所有移动终端的卸载决策与资源优化问题通过博弈模型进行表示并引入势博弈概念证明该模型具有纳什均衡性。随后在已知终端卸载决策的基础上,对上、下行频谱资源以及边缘服务器的计算资源进行联合优化并利用拉格朗日乘子法获取计算资源与频谱资源的最优解。最后,基于获得的最优解,利用迭代法得到所有移动终端的纳什均衡解。仿真结果表明,本文所提出的基于博弈论的计算卸载与资源分配算法与现有算法相比可以获取更低的时延,进而提升用户体验。

1 基于 MEC 的系统建模与问题描述

1.1 系统模型

MEC 系统的网络框架如图 1 所示,该系统由 1 台边缘服务器与 1 个基站组成。

边缘服务器与基站可以为其覆盖范围内的 N 个移动终端提供计算与通信服务。其中多个移动终



端的集合表示为 $L = \{i : i = 1, 2, \dots, N\}$, 单个终端由 i 进行表示。为了防止终端间干扰, 基站的频谱资源通过正交的方式分配给每个终端且总频谱带宽为 B 。假设每个终端只执行一个计算任务 $w_i = \{s_i, m_i, c_i\}$, 其中 s_i 表示输入数据的大小, m_i 表示计算结果数据的大小, c_i 表示一个任务所需中央处理器 (central processing unit, CPU) 循环总量。由于计算任务既可以本地执行也可以在边缘服务器执行, 因此移动终端 i 的卸载决策由 a_i 进行表示, 其中 $A_i \in \{0, 1\}$ 且 $a_i \in A_i$ 。当 $A_i = 1$ 时, 表示终端 i 将会选择将任务卸载至 MEC 服务器。 $A_i = 0$ 时, 则表示任务在终端 i 执行。因此所有终端的卸载策略表示为 $A = \{A_i | A_i \in \{0, 1\}, i \in L\}$ 。

1.2 通信模型

由于每个移动终端在执行任务时将会占用上行与下行的频谱资源, 因此终端 i 的上行传输速率可以表示为

$$r_i^u = k_i B \log\left(1 + \frac{|h_{i,B}|^2 P_{i,B} d^{-r}}{\sigma^2}\right) \quad (1)$$

其中, k_i 为终端上传数据所占上行带宽百分比, $P_{i,B}$ 表示终端的发送功率, $h_{i,B}$ 表示基站与终端间的信道衰落系数, d 为终端与基站的距离, r 为路径损耗, σ^2 为信道的噪声功率。同理下行传输速率表示为

$$r_i^d = \xi_i B \log\left(1 + \frac{|h_{B,i}|^2 P_B d^{-r}}{\sigma^2}\right) \quad (2)$$

其中, ξ_i 为终端接收下行数据所占带宽百分比, $h_{B,i}$ 表示为基站与终端间的信道衰落系数, P_B 为基站的发射功率。

1.3 计算模型

当移动终端将任务卸载至边缘服务器后, 此时

的任务时延主要由上行传输时延 $t_{i,u}$ 、下行传输时延 $t_{i,d}$ 、服务器计算时延 $t_{i,m}$ 以及回程链路时延 $t_{i,b}$ 4 部分组成。由于基站与服务器之间是通过有线的方式进行连接,因此回程链路时延忽略不计。此外当移动终端选择在边缘服务器执行任务时,分配给移动终端的计算资源不能高于边缘服务器最大计算能力,其中边缘服务器的最大计算能力用 f_m 表示。

(1) 本地执行任务。当任务 w_i 在本地执行时,计算时延表示为

$$t_{i,l} = \frac{c_i}{f_{i,l}} \quad (3)$$

其中, $f_{i,l}$ 为移动终端的计算能力。因此终端 i 在本地执行任务的总时延表示为

$$U_{i,l} = t_{i,l} \quad (4)$$

(2) 边缘服务器执行任务。当计算任务卸载至边缘服务器时,其上行数据传输时延表示为 $t_{i,u} = \frac{s_i}{r_i^u}$, 下行数据接收时延表示为 $t_{i,d} = \frac{m_i}{r_i^d}$, 服务器计算时延表示为 $t_{i,m} = \frac{c_i}{f_i}$, 其中 f_i 表示为服务器为终端 i 分配的计算资源。因此移动终端 i 将任务卸载至服务器产生的总时延表示为

$$U_{i,m} = t_{i,u} + t_{i,d} + t_{i,m} \quad (5)$$

1.4 问题形式化描述

由于不同任务所需的资源不同,为了满足所有移动终端的任务时延要求,在计算与通信资源约束条件下对终端的计算卸载决策 A , 上行频谱资源 K , 下行频谱资源 P 以及服务器计算资源 F 4 个方面联合优化,其中优化目标可以表示为

$$\min_{A,F,K,P} Z(a) = \sum_{i=1}^N U_{i,m} a_i + U_{i,l} (1 - a_i) \quad (6)$$

$$\text{s. t. } \text{C1: } \sum_i f_i \leq f_m \quad \forall i \in L$$

$$\text{C2: } \sum_i k_i \leq 1 \quad \forall i \in L$$

$$\text{C3: } \sum_i \xi_i \leq 1 \quad \forall i \in L$$

$$\text{C4: } f_{i,l} \geq 0 \quad \forall i \in L$$

$$\text{C5: } a_i \in \{0,1\} \quad \forall i \in L$$

其中 $F = \{f_{i,m}: \sum_i f_{i,m} \leq f_m, \forall i \in L\}$, $a = (a_1, a_2, \dots, a_n)$, $K = \{k_i: \sum_i k_i \leq 1, \forall i \in L\}$, $P = \{\xi_i:$

$\sum_i \xi_i \leq 1, \forall i \in L\}$ 。其中限制条件 C1 表示分配给所有移动终端计算资源总和小于等于服务器的计算能力,限制条件 C2、C3 表示分配给所有移动终端的上行与下行频谱资源的百分比,限制条件 C4 则表示终端的本地计算资源是非负的,C5 表示为移动终端的卸载决策。此外由式(6)可以看出,由于 a_i 为整型变量,因此问题式(6)为混合整型线性规划问题且为非凸的,因此很难得到直解。为了解决以上问题,本文提出了一种基于博弈论的计算卸载与资源分配算法。

2 基于博弈论的计算卸载与资源分配算法

博弈论是指博弈或者对弈过程中,通过考虑或预测参与人的实际行为从而优化参与者的决策。为了获取最优卸载策略,接下来将把上述问题建模为博弈模型。

2.1 博弈形式化表述

首先将问题式(6)通过博弈模型 $G = \{L, (A_i)_{i \in L}, W_i\}$ 来表示,其中 L 为终端个数, A_i 为终端的卸载策略, w_i 为终端 i 的时延效用函数。 $a_{-i} = (a_1, a_2, \dots, a_{i-1}, a_{i+1}, \dots, a_N)$ 表示除了终端 i 以外其他终端的卸载决策。由于每个移动终端用户具有自私性,因此为了最小化所有终端的任务时延,所有参与者会对有限的计算资源、频谱资源进行竞争。因此在竞争条件下,每个移动终端的时延表示为

$$\min_{a_i \in [0,1]} W_i(a_i, a_{-i}) = U_{i,m} a_i + U_{i,l} (1 - a_i) \quad (7)$$

由于该模型具有纳什均衡的特性,即存在最优的卸载策略使得每个终端时延最低。因此在所有移动终端完成决策后,需要对卸载至服务器的移动设备进行通信与计算资源优化,而资源的优化目标可以表示为

$$\begin{aligned} \min_{k, \xi, f} W(k, \xi, f) &= \sum_{i=1}^N \frac{s_i}{k_i B \log_2 \left(1 + \frac{|h_{i,B}|^2 p_{i,B} d^{-r}}{\sigma^2} \right)} \\ &+ \frac{m_i}{\xi_i B \log_2 \left(1 + \frac{|h_{B,i}|^2 p_B d^{-r}}{\sigma^2} \right)} + \frac{c_i}{f_i} \end{aligned} \quad (8)$$

根据海塞矩阵很容易证明函数 $W(k, \xi, f)$ 为凸函数, 此外由于限制条件 C1 ~ C3 是线性的, 因此问题式(8)变为凸优化问题^[20]。为了解决该问题, 采用对偶的方法, 相应的对偶函数表示为

$$\begin{aligned} & \max_{\mu, \lambda, \theta} \min_{k, \xi, f} L(\mu, \lambda, \theta, k, \xi, f) \\ &= \sum_{i=1}^N \frac{s_i}{k_i B \log_2 \left(1 + \frac{|h_{i,B}|^2 p_{i,B} d^{-r}}{\sigma^2} \right)} \\ &+ \frac{m_i}{\xi_i B \log_2 \left(1 + \frac{|h_{B,i}|^2 p_B d^{-r}}{\sigma^2} \right)} + \frac{c_i}{f_{i,m}} \\ &+ \lambda \left(\sum_{i=1}^N k_i - 1 \right) + \theta \left(\sum_{i=1}^N \xi_i - 1 \right) + \mu \left(\sum_{i=1}^N f_i - f_M \right) \end{aligned} \quad (9)$$

可以看出式(9)分为 2 个部分。第 1 部分为内层函数, 主要由变量 k, f, ξ 组成。第 2 部分为外层函数, 其中变量 μ, λ, θ 为拉格朗日乘子且为非负值。由于内层函数是凸的, 因此将通过卡罗需-库恩-塔克 (Karush-Kuhn-Tucker, KKT) 条件获取最优解。为了表示更加直观, 将上、下行传输速率表示为

$$\begin{aligned} H_i &= B \log_2 \left(1 + \frac{|h_{i,B}|^2 p_{i,B} d^{-r}}{\sigma^2} \right) \\ Q_i &= B \log_2 \left(1 + \frac{|h_{B,i}|^2 p_B d^{-r}}{\sigma^2} \right) \end{aligned}$$

因此最优解 k_i^*, ξ_i^*, f_i^* 可以表示为

$$k_i^* = \sqrt{\frac{s_i}{\lambda H_i}} \quad (10)$$

$$\xi_i^* = \sqrt{\frac{m_i}{\theta Q_i}} \quad (11)$$

$$f_i^* = \sqrt{\frac{c_i}{\mu}} \quad (12)$$

为了优化外层函数, 在已知内层函数最优解的基础上通过梯度法对拉格朗日乘子进行优化, 其中拉格朗日乘子表示如下:

$$\mu(t+1) = [\mu(t) - \partial_1(t)(f_M - \sum_i f_i)]^+ \quad (13)$$

$$\lambda(t+1) = [\lambda(t) - \partial_2(t)(1 - \sum_i k_i)]^+ \quad (14)$$

$$\theta(t+1) = [\theta(t) - \partial_3(t)(1 - \sum_i \xi_i)]^+ \quad (15)$$

其中, $\partial_1, \partial_2, \partial_3$ 为迭代步长, t 为迭代次数且 $t \geq 0$ 。

2.2 纳什均衡的存在性

为了证明博弈模型 G 存在一个纯策略纳什均衡, 本文引入势博弈的概念。

定义 1 针对博弈模型 $G = \{L, (A_i)_{i \in L}, W_i\}$, 如果存在一组策略 $a^* = (a_1^*, a_2^*, \dots, a_n^*)$ 是纳什均衡的, 那么在其他终端策略不变的情况下, 当前移动终端不能单方面改变自己的卸载策略而降低自己的时延效用函数。即:

$$W_i(a_i^*, a_{-i}^*) \leq W_i(a_i, a_{-i}^*), \forall a_i \in \{0, 1\}, i \in L \quad (16)$$

定义 2 如果存在一个势函数 $\varphi(a)$ 对于任何一个终端设备 i 在 $i \in La_i, a_i' \in A_i$ 条件下使得下式成立:

$$\begin{aligned} W_i(a_i, a_{-i}) - W_i(a_i', a_{-i}) &= \varphi(a_i, a_{-i}) - \varphi(a_i', a_{-i}) \end{aligned} \quad (17)$$

则该博弈为完全势博弈。由于普通势博弈包含完全势博弈, 因此完全势博弈具有普通势博弈的所有属性。

定理 1 如果一个普通势博弈具有有限的策略集合, 那么它存在一个纯策略的纳什均衡。

定理 2 如果一个博弈模型是完全势博弈且存在势函数, 那么存在一个纯策略的纳什均衡且具有有限递增属性。

证明 首先设置势函数为

$$\varphi(a) = \sum_{i=1}^N a_i U_{i,m} + (1 - a_i) U_{i,l} \quad (18)$$

随后设置 2 个不同策略 $a = (a_i, a_{-i})$, $a' = (a_i', a_{-i})$ 并将 2 个不同的策略带入函数 $\varphi(a)$ 与 $w_i(a_i, a_{-i})$ 中得到:

$$\begin{aligned} \varphi(a) &= a_i U_{i,m} + (1 - a_i) U_{i,l} + \sum_{j \neq i}^N a_j U_{j,m} \\ &+ (1 - a_j) U_{j,l} \end{aligned} \quad (19)$$

$$\begin{aligned} \varphi(a') &= a'_i U_{i,m} + (1 - a'_i) U_{i,l} + \sum_{j \neq i}^N a'_j U_{j,m} \\ &+ (1 - a'_j) U_{j,l} \end{aligned} \quad (20)$$

$$W_i(a) = a_i U_{i,m} + (1 - a_i) U_{i,l} \quad (21)$$

$$W_i(a') = a_i' U_{i,m} + (1 - a_i') U_{i,l} \quad (22)$$

基于式(19)~(22)可以得到:

$$\varphi(a) - \varphi(a') = W_i(a) - W_i(a') \quad (23)$$

因此上述博弈模型为完全势博弈且存在纯策略纳什均衡。

2.3 算法描述

由于边缘服务器拥有所有接入终端的完整信息,因此服务器可以根据接入信息获取优化后的卸载策略并将该策略发送至移动终端。根据纳什均衡性,所有移动终端将会服从该卸载决策。详细的算法描述如算法 1 所示。

算法 1 基于博奕论的计算卸载与资源分配算法

步骤 1 初始话:对终端卸载策略 a ,终端发射功率 $p_{i,B}$,
 p_B ,任务的输入数据大小 s_i 、输出数据大小 m_i 以及任务
 所需的计算资源 c_i 进行初始化;

步骤 2 在已知卸载策略基础上,获取任务卸载至本地终
 端时所需的执行时延 $U_{i,l}$;

步骤 3 对于卸载至云端的任务通过式(10)~(12)获取
 上、下行资源以及计算资源的最优解 k_i^*, ξ_i^*, f_i^* ;

步骤 4 在步骤 3 的最优解基础上,通过式(13)~(15)
 对拉格朗日乘子进行更新;

步骤 5 基于步骤 3 和 4,将卸载至服务端的任务时延
 $U_{i,m}$ 并与卸载至本地的任务时延 $U_{i,l}$ 进行比较。如果
 $U_{i,m} > U_{i,l}$,则终端卸载至本地即 $a_i = 0$ 否则 $a_i = 1$ 。通过
 循环获取新的计算卸载策略 a ;

步骤 6 在已知策略 a 的基础上,设定其他终端卸载策略
 不变,单一改变终端 i 的卸载策略即 $a_i = 1$,然后通过步
 骤 3、4 获取终端的任务时延。此时,如果 $U_{i,l} > U_{i,m}$, a_i
 $= 1$ 同时对策略 a 进行更新。然后 $i+1$ 且 $a_i = 1$,重复步
 骤 6 直到终端 i 等于 N 时循环结束。此时得到最优的卸
 载策略 a 以及资源分配 k_i^*, ξ_i^*, f_i^* 。

3 仿真结果与分析

为了验证本文所提基于博奕论的计算卸载与资
 源分配算法(CORAG)性能,在信道条件不变的基础
 上,分别与文献[21]所提的本地卸载算法(LOC)、
 随机卸载算法(random offloading completely, ROC)
 以及文献[19]提出的计算卸载与上行频谱资源分
 配(computation offloading and uplink resource game,

COURG)进行比较分析。

3.1 参数设置

在仿真实验中,假设有 2~20 个移动设备随机分布
 在基站覆盖范围为 100 m 的区域内。 $h_{i,B}$ 与 $h_{B,i}$
 服从均值为 1 的指数分布; $f_{i,l}$ 表示为移动终端自身的
 计算能力并服从高斯分布 $f_{i,l} \sim CN(\mu_1, \delta_1^2)$, 其中
 $\mu_1 = 500$ Megacycles/s, $\delta_1^2 = 50$; 而计算任务数据与
 计算结果数据的大小分别表示为 s_i 、 m_i , 并服从高
 斯分布 $s_i \sim CN(\mu_2, \delta_2^2)$, $m_i \sim CN(\mu_3, \delta_3^2)$, 其中
 $\mu_2 = \mu_3 = 500$ kB, $\delta_2^2 = \delta_3^2 = 50$ 。此外任务的计算负载
 c_i 同样服从高斯分布 $c_i \sim CN(\mu_4, \delta_4^2)$, $\mu_4 = 1000$
 Megacycles/s, $\delta_4^2 = 50$ 。其余的系统参数如表 1 所
 示。

表 1 仿真参数设置

参数	值
上下行链路带宽 (B)	20 MHz
基站发射功率 (p_B)	27 dBm
移动终端发射功率 ($p_{i,B}$)	23 dBm
路径损耗 (r)	4
噪声功率 (δ^2)	-174 dBm
边缘服务的计算能力 (f_M)	30 000 Megacycles/s

3.2 仿真分析

首先考虑移动终端个数变化对系统总时延的影
 响。当终端的发射功率为 23 dBm,边缘服务器的计
 算能力为 30 000 Megacycles/s,上下行链路带宽相同
 且为 20 MHz 时,仿真结果如图 2 所示。图 2 表明,
 本文提出的 CORAG 算法的性能优于 LOC、ROC 以
 及 COURG 算法,随着移动终端数量增加 CORAG 算
 法优势更加明显。这是因为在计算与通信资源有限
 的条件下,CORAG 算法可以更好地为不同的移动终
 端分配最优的计算与通信资源,因此所有移动终端
 的总时延最低。另外从图 2 可以看出,当所有移动
 设备的计算任务在本地执行时,由于计算任务时延
 的大小只与自身计算能力有关,因此随着终端数的
 增加,LOC 算法的总时延是线性递增的;当采用随
 机卸载策略时,由于一部分移动终端会将任务卸
 载至边缘服务器而另一部分移动终端在本地执行计
 算任务,因此所有移动终端总时延低于 LOC 算法且呈

波动变化。此外,由于 COURG 算法忽略了下行资源的优化,因此其总时延略高于 CORAG 算法。

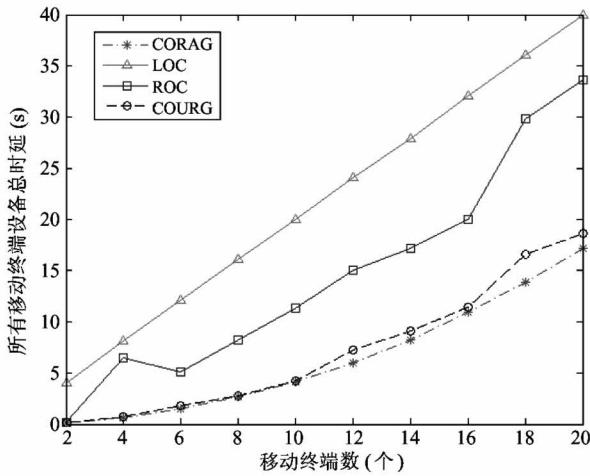


图 2 不同算法在不同移动终端数量下的总时延比较

当移动终端数量为 10、边缘服务器计算能力为 30 000 Megacycles/s 且上、下行带宽为 20 MHz 时,比较在不同输入数据大小下不同算法的性能,仿真结果如图 3 所示。由图 3 可以看出,随着上行输入数据的增大,CORAG 算法总时延低于 LOC、ROC 以及 COURG 算法。这是因为 CORAG 算法可以根据获得输入数据的大小分配最优的上、下行带宽资源以及计算资源,进而使得所有终端设备的总时延最低。另外从图 3 明显看出,当计算任务在本地执行时,总时延仅与任务所需计算资源有关,因此 LOC 算法没有明显变化。

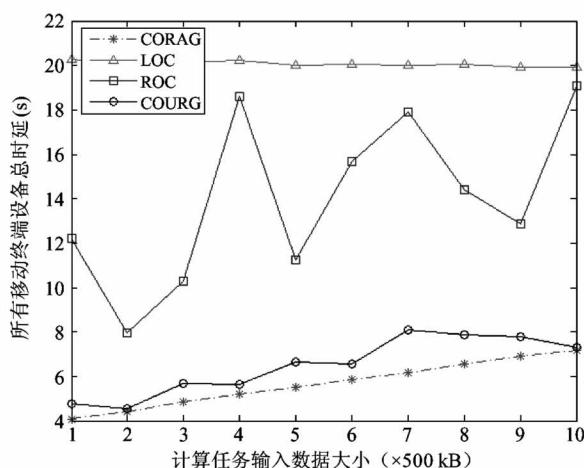


图 3 不同算法在不同输入数据大小下的总时延比较

图 4 表示不同算法在不同计算结果数据大小下的性能仿真分析。仿真结果表明,随着计算结果数据的增大,CORAG 算法的总时延低于 LOC、ROC 以及 COURG 算法。由图 4 可以看出,当计算结果大小不变时,所提算法性能明显优于其他 3 种算法且随着计算结果数据的逐渐增大,CORAG 算法性能优势明显。此外,由于 COURG 算法忽略了下行带宽的优化,因此随着计算结果数据的增大,所有移动终端的总时延波动较大。而 LOC 算法的计算任务时延与计算结果大小无关,因此所有设备的总时延基本不变。最后,由于 ROC 算法中任务总是随机进行卸载,因此曲线是波动变化的。

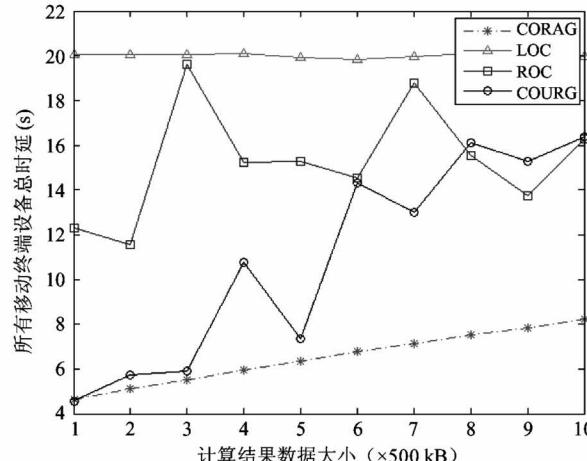


图 4 不同算法在不同计算结果数据大小下的总时延比较

为了进一步比较本文所提 CORAG 算法与 LOC、ROC 以及 COURG 3 种算法的性能,当移动终端数量为 10、移动终端的计算能力为 1 000 Megacycles/s 且边缘服务器计算能力为 35 000 Megacycles/s 时,对不同算法在任务所需的 CPU 循环变化下进行仿真分析,仿真结果如图 5 所示。仿真结果表明,随着任务所需 CPU 循环的增加,消耗的资源逐渐递增。因此在资源有限条件下,4 种算法的总时延逐渐上升。此外,由于 CORAG 与 COURG 算法对计算与通信资源进行了优化,因此算法性能优于 LOC 与 ROC 且上升趋势较为缓慢。另外从图 5 可以看出,随着任务所需 CPU 循环逐步增加,COURG 算法曲线略微波动且在性能上与 CORAG 算法相近。这是因为随着任务所需 CPU 循环数的增加 COURG 算法总能为移动终端分配最优的计算资源,但由于该算法忽

略了下行频谱资源的优化,因此 COURG 曲线会略微波动。

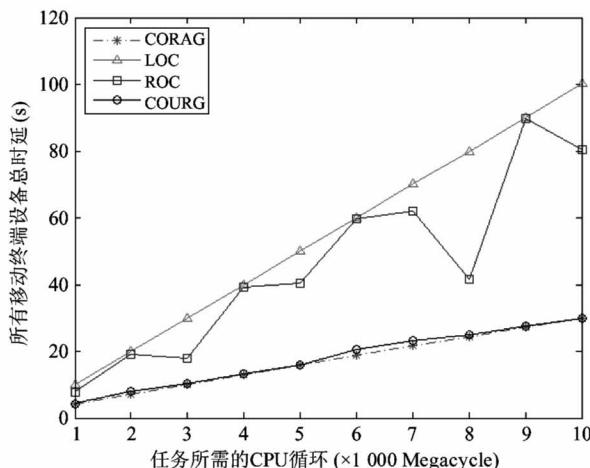


图 5 不同算法在不同任务所需的 CPU 循环变化下总时延比较

最后,在不同边缘服务器的计算能力条件下对 CORAG 算法进行仿真,其仿真结果如图 6 所示。仿真结果表明,当移动终端数低于 4 时,边缘服务器计算能力大小对所有移动终端的总时延影响较小;当移动终端数从 6 逐步递增至 20 时,边缘服务器计算能力大小对所有移动终端的总时延影响较大。另外,通过对比计算能力为 30 000 Megacycles/s 与 50 000 Megacycles/s 曲线可知,当移动终端个数由 14 增长至 20 时,2 条曲线总时延的差值也在逐步增大。

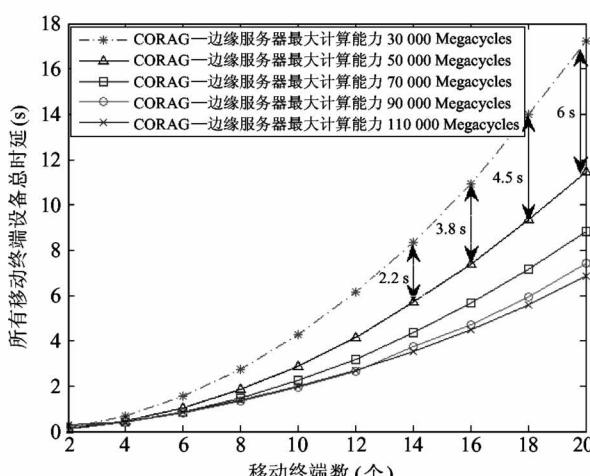


图 6 随着移动终端数增加,不同边缘服务器计算能力条件下的 CORAG 算法性能比较

因此从以上结果可以得出,边缘服务器的计算能力越高,移动终端的时延越低。这是因为当终端个数较少时,现有的计算资源可以满足终端用户的需求;然而随着终端数量的增加其需要的计算资源将越来越多,此时服务器的计算资源越丰富,移动终端的总时延越低,算法的性能越优越。

4 结 论

本文针对计算能力不足的移动终端处理低时延、高可靠应用而产生高时延的问题,提出了一种基于博弈论的计算卸载与资源分配联合优化算法。该算法在设计过程中首先将计算卸载与资源分配问题表示为非合作博弈模型并通过引入势博弈概念证明该博弈模型存在纳什均衡状态;其次在已知移动终端卸载的决策条件下,利用拉格朗日乘子法获取计算资源以及上下行频谱资源的最优解;最后在最优解的基础上,采用迭代法获得移动终端的纳什均衡解。仿真结果表明,与 LOC 算法、ROC 算法以及 COURG 算法相比,本文提出的 CORAG 算法可以降低所有移动终端的任务时延,提升终端用户的服体验。本文主要研究是在单基站覆盖下多个移动终端在非合作场景中的时延优化问题。然而在实际应用场景中,移动终端是频繁移动的,而这将增加不同终端间的连接几率并使移动终端间的资源复用成为可能;此外由于网络具有不同的负载特性,如何根据不同网络特点进行有效的资源分配也是未来研究方向^[22,23]。因此在终端频繁移动的条件下根据不同的网络特点,提出有效的资源分配算法降低终端任务时延是下一步的研究重点。

参考文献

- [1] Mach P, Becvar Z. Mobile edge computing: a survey on architecture and computation offloading[J]. *IEEE Communications Surveys and Tutorials*, 2017, 19(3): 1628-1656
- [2] Zhou Y, Tian L, Liu L, et al. Fog computing enabled future mobile communication networks: a convergence of communication and computing[J]. *IEEE Communications Magazine*, 2019, 53(5): 20-27

- [3] Liu L, Zhou Y, Yuan J, et al. Economically optimal MS association for multimedia content delivery in cache-enabled heterogeneous cloud radio access networks [J]. *IEEE Journal on Selected Areas in Communications*, 2019, 37(7) : 1584-1593
- [4] Liu L, Zhou Y, Zhuang W, et al. Tractable coverage analysis for hexagonal macrocell-based heterogeneous UDNs with adaptive interference aware CoMP[J]. *IEEE Transactions on Wireless Communications*, 2019, 18(1) : 503-517
- [5] Zheng K, Zheng Q, Chatzimisios P, et al. Heterogeneous vehicular networking: a survey on architecture, challenges and solutions[J]. *IEEE Communications Surveys and Tutorials*, 2015, 17(4) :2377-2396
- [6] Lei L, Zhong Z, Zheng K, et al. Challenges on wireless heterogeneous networks for mobile cloud computing[J]. *IEEE Wireless Communications*, 2013, 20(3) :34-44
- [7] Zhang Y, Liu H, Jiao L, et al. To offload or not to offload: an efficient code partition algorithm for mobile cloud computing[C] // IEEE International Conference on Cloud Networking (CloudNet), Paris, France, 2012: 80-86
- [8] Fernando N, Loke S W, Rahayu W. Mobile cloud computing: a survey [J]. *Future Generation Computer Systems*, 2013, 29(1) :84-106
- [9] Li J R, Li X Y, Gao Y Q, et al. Energy saving research on mobile cloud computing in 5G[J]. *Chinese Journal of Computers*, 2017,40(7) : 1491-1516
- [10] Zohreh S, Saeid A, Abdullah G, et al. Heterogeneity in mobile cloud computing: taxonomy and open challenges [J]. *IEEE Communications Surveys and Tutorials*, 2013, 16 (1) : 369-392
- [11] European Telecommunications Standards Institute (ETSI). ETSI-GS-MEC-002-V1. 1. 1 Mobile Edge Computing (MEC): Technical Requirements [S]. Sophia Antipolis Cedex, France: European Telecommunications Standards Institute, 2016
- [12] 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
- [13] Guo S, Xiao B, Yang Y, et al. Energy-efficient dynamic offloading and resource scheduling in mobile cloud computing[C] // IEEE International Conference on Computer Communications (INFOCOM), San Francisco, USA, 2016: 1-9
- [14] Zhang X, Mao Y, Zhang J, et al. Multi-objective resource allocation for mobile edge computing systems[C] // IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Montreal, Canada, 2017: 1-5
- [15] 余韵, 连晓灿, 朱宇航, 谭国平, 等. 增强现实场景下移动边缘计算资源分配优化方法[J]. 计算机应用, 2019, 39(1) :22-25
- [16] Chen M H, Dong M, Liang B. Multi-user mobile cloud offloading game with computing access point[C] // IEEE International Conference on Cloud Networking (Cloudnet), Pisa, Italy, 2016: 64-69
- [17] 林涛, 秦冬阳, 马同宽, 等. 基于博弈论的移动边缘计算任务调度研究[J]. 计算机仿真, 2018 , 35 (11) : 399-403
- [18] Chen X. Decentralized computation offloading game for mobile cloud computing[J]. *IEEE Transactions on Parallel and Distributed Systems*, 2015, 26(4) :974-983
- [19] Zhang J, Xia W, Zhang Y, et al. Joint offloading and resource allocation optimization for mobile edge computing [C] // IEEE Global Communications Conference (Globecom), Singapore, 2017: 1-6
- [20] Boyd S, Vandenberghe L. Convex Optimization [M]. Cambridge, UK: Cambridge University Press, 2004
- [21] Zhang Y, Niyato D, Wang P. Offloading in mobile cloudlet systems with intermittent connectivity [J]. *IEEE Transactions on Mobile Computing*, 2015, 14(12) :2516-2529
- [22] Liu L, Zhou Y, Garcia V, et al. Load aware joint CoMP clustering and inter-cell resource scheduling in heterogeneous ultra dense cellular networks[J]. *IEEE Transactions on Vehicular Technology*, 2018 , 67(3) : 2741-2755
- [23] Liu L, Zhou Y, Tian L, et al. CPC-based backward compatible network access for LTE cognitive radio cellular networks[J]. *IEEE Communication Magazine*, 2015 , 53 (7) : 93-99

Joint optimization strategy of computation offloading and resource allocation in mobile edge computing

Long Long^{* ***}, Liu Zichen^{*}, Shi Jinglin^{* ***}, Zhou Yiqing^{* ***}, Qiu Dawei^{* ***}, Xu Shunqing^{* ***}

(^{*} Beijing Key Laboratory of Mobile Computing and Pervasive Device, Institute of Computing Technology,
Chinese Academy of Sciences, Beijing 100190)

(^{**} University of Chinese Academy of Sciences, Beijing 100049)

Abstract

In order to reduce high delay caused by mobile terminals with insufficient computing power in processing low latency applications in mobile edge computing (MEC), a joint optimization strategy based on game theory for computational offloading and resource allocation is proposed. Firstly, the offloading decision and resource optimization problem of multiple mobile terminals is formulated as a non-cooperative game model and the existence of Nash equilibrium is proved. Then, based on the offloading decision of the mobile terminals, a Lagrange multiplier method is used to obtain optimal computation resource, uplink and downlink spectrum resource. Finally, the iterative method is used to obtain the Nash equilibrium solution. The simulation results show that the proposed algorithm can minimize the task delay of mobile terminals and improve user satisfaction with the constraint of resource.

Key words: mobile edge computing (MEC), game theory, resource allocation, computation offloading