

DOI:10.19322/j.cnki.issn.1006-4710.2021.04.014

# 云机器人系统的计算卸载策略研究

孙 弋, 许志杰

(西安科技大学 通信与信息工程学院, 陕西 西安 710054)

**摘要:** 为了解决云机器人系统中的计算卸载问题,提高边缘云资源的使用效率,降低群组机器人的能量消耗和对硬件计算资源的要求,提升机器人服务效率和质量,本文设计一种基于 Stackelberg 博弈的云机器人计算卸载策略。该方法将云机器人竞争同一边缘“网络”中相同类型资源或服务的过程抽象为主从关系的博弈问题,寻求机器人终端进行计算卸载需支付的成本开销与边缘云服务提供者的效用收益之间的平衡。边缘云服务器属于领导者,通过对自身计算能力的定价实现有限计算资源的收益最大化。每个移动机器人终端根据边缘云的定价策略,做出计算卸载的最佳决策,使其计算卸载策略达到 Nash 均衡状态,最大化自身效用。本文通过对云机器人业务进行任务分割,实现计算任务的部分卸载,证明了基于 Stackelberg 博弈模型的云机器人计算卸载系统的正确性,提出了一种结合 Hamming 距离的动态规划算法求解博弈的最终任务卸载策略。实验结果表明该算法降低了云机器人系统中的局部计算能耗,缩短了平均任务完成时间。

**关键词:** 云机器人; 计算卸载; Stackelberg 博弈; 动态规划

**中图分类号:** TP242.6

**文献标志码:** A

**文章编号:** 1006-4710(2021)04-0562-08

## Research on computing offloading strategy for cloud robot system

SUN Yi, XU Zhijie

(College of Communication and Information Engineering, Xi'an University of Science and Technology, Xi'an 710054, China)

**Abstract:** In order to solve the problem of computing offloading in cloud robot system, improve the utilization efficiency of edge cloud resources, reduce the energy consumption of swarm robots and the requirements for hardware computing resources, and improve the efficiency and quality of robot service, this paper designs a computing offloading strategy based on the Stackelberg game in Cloud Robotics. This method abstracts the process of cloud robots competing for the same type of resources or services in the same edge from the game problem of master-slave relationship, and seeks the balance between the cost of robot terminal computing and the utility income of edge cloud service providers. Edge cloud server is a leader, and it can maximize the income of limited computing resources by pricing its own computing power; according to the pricing strategy for edge cloud, each mobile robot makes the best decision on the computing offloading strategy, making its computing offload strategy reach Nash equilibrium and maximize its own utility. In this paper, by dividing the tasks of Cloud Robotics, the partial offloading of computing tasks is realized, which proves the correctness of the cloud robot computing offloading system based on the Stackelberg game model and proposes a dynamic programming algorithm combined with Hamming distance to solve the final task offloading strategy for the game. Experimental results show that the algorithm reduces the local computing energy consumption and the average task completion time in the cloud robot system.

**Key words:** cloud robot; computation offloading; Stackelberg game theory; dynamic programming

**收稿日期:** 2020-03-15; **网络出版日期:** 2021-07-14

**网络出版地址:** <https://kns.cnki.net/kcms/detail/61.1294.n.20210713.1755.002.html>

**第一作者:** 孙弋,男,博士,教授,研究方向为嵌入式系统、信号与信息处理。E-mail: 409298131@qq.com

**通信作者:** 许志杰,男,硕士生,研究方向为云机器人系统。E-mail: Kevinxu653@163.com

近年来伴随着嵌入式系统性能的提升、传感器技术的蓬勃发展<sup>[1]</sup>以及移动数字网络技术的快速迭代,全球范围内,应用机器人市场规模增长明显<sup>[2]</sup>。不同于执行特定程序的物理设备,机器人具有高度的自主决策能力,对不同环境和场景适应性强,且机器人群组之间的协作愈发成熟<sup>[3]</sup>。如何实现机器人集群的大规模应用以及使机器人在有限功耗的限制下,具备强大的计算能力和低廉的部署成本成为亟需解决的热点需求。基于网络技术将机器人本体需完成的数据分析与处理、行为决策与规划、机器人群组协同等复杂的计算任务卸载到云端执行,通过数据共享,凭借云端专有的软硬件资源代替机器人执行更加复杂的计算任务、存储更大规模数据、为机器人提供服务的系统称为云机器人系统架构。郭宇<sup>[1]</sup>利用连通支配集对云机器人底层网络建模,并结合能耗敏感模型实现对整个机器人自组织网络中的能量和时间延迟进行分析,并在该模型下设计了基于遗传算法的计算卸载策略。博弈论是研究分布式多智能体系统中决策者之间相互作用关系的数学工具,而云机器人系统作为典型的分布式系统,其服务的提供者与使用者,及使用者与使用者之间的博弈时刻存在,其中边缘云拥有对计算资源的定价权,是博弈的领导者,而云机器人依据自身情况及所需资源价格做出卸载决策,是博弈的跟随者。Xu等<sup>[4]</sup>对云机器人系统中的任务进行分割与建模,基于多边缘网络中云机器人计算卸载问题设计了改进的博弈论算法,提高机器人网络服务质量和用户体验。尤斐然<sup>[5]</sup>应用博弈论设计资源优化分配的策略,分析一对多、多对多、多层联合博弈的模型下的资源分配效率。上述文献从博弈论的角度对资源分配与能耗进行分析,但未讨论博弈各方决策时的相互影响。针对上述不足,本文将多个机器人终端竞争同一边缘云服务器中相同类型资源或服务的行为建模为 Stackelberg 博弈模型,基于逆向归纳法求解在多阶段博弈过程中边缘云与机器人终端的决策问题,求得云机器人系统博弈中双方策略最优的 Nash 均衡解,实现双方效用函数的最大化。

## 1 相关研究

### 1.1 云机器人系统架构

云机器人系统是在传统的分布式网络机器人系统之上,结合云计算<sup>[6]</sup>服务发展而来的一种新兴发展方向。这一概念在 2010 年由 Kuffner 教授<sup>[7]</sup>首次提出,旨在利用互联网大规模并行计算和近乎无限的分布式存储能力为机器人设计统一的服务交互模型,扩展其计算能力以及增强其服务能力。根据

服务提供方式的不同,将该模型分为面向服务(SOA)的平台架构和提供机器人即服务(RaaS)的平台架构<sup>[8]</sup>。Arumugam等<sup>[9]</sup>基于平台即服务的思想,设计 DAVinCi 的云机器人计算框架,并采用 Map/Reduce 的方式实现了机器人同步定位与地图构建算法的云服务云平台。Quintas等<sup>[10]</sup>提出了面向服务的机器人系统,其中机器人和智能空间共享知识。Du等<sup>[11]</sup>基于机器人即服务提出了机器人云中心(robot cloud center)的框架,其结合了 SOA 架构和云计算模型,使任务以最大效率,耗费最少资源完成。结合相关研究背景及云计算的特点,本文拟将机器人任务封装为众多独立微服务,构建 RaaS 的云机器人服务管理系统,机器人通过端到云的网络向云端请求所需计算服务。计算卸载策略是云机器人进行计算卸载的关键,其决定任务是否进行卸载以及卸载数量。

### 1.2 计算卸载

计算卸载<sup>[12-13]</sup>是解决边缘计算中用户终端设备在计算和存储资源等方面的不足,是云机器人架构最重要的特点之一。用户终端将对时间延迟不敏感的计算密集型任务迁移到专用服务器上运算,待计算完成后接收计算结果的过程称为计算卸载,其流程见图 1。

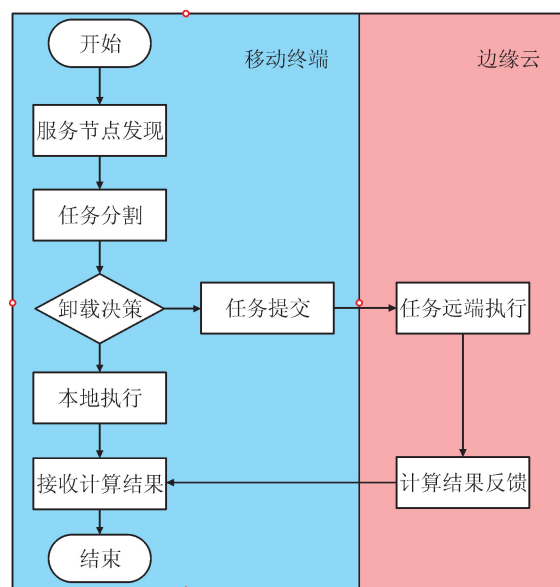


图 1 计算卸载流程图

Fig. 1 Calculation offloading flow chart

计算卸载分二进制卸载和部分卸载两种方式。二进制卸载方式指计算任务不可分割,其程序和数据作为统一整体卸载到边缘云执行,由此其对通信网络的传输能力、网络延迟提出了较高的要求,服务作为整体卸载到边缘云端,降低了云端资源调度策略的设计难度,但使请求服务调度失效时的异常处理难度增加。

Chen 等<sup>[14]</sup>研究在任务执行时延和能耗约束下,多参与者通过非合作博弈达到最优计算策略的二进制卸载问题。Ma 等<sup>[15]</sup>研究无线网络组网环境中多用户终端的计算卸载问题,结合博弈论将所有用户的计算任务,无线频带分配和计算资源进行建模,获得使任务执行成本最小的最优策略。Chen 等<sup>[16]</sup>也是利用非合作博弈模型分析多用户终端之间卸载策略的相互影响,基于降低任务执行总代价和时间延迟,获得最优的计算卸载策略使通信与计算资源分配最优。部分卸载方式指将计算任务分割为多个子任务,一部分在本地执行,一部分在边缘云执行,既减少计算任务传输数据总量,又提高子任务执行的并行性,充分发挥本地和边缘侧协同计算的优势,提高任务执行的效率,同时需为到达边缘侧的子任务设计任务调度机制,满足依赖关系的子任务串行执行,相互独立的子任务并行执行。Dai 等<sup>[17]</sup>基于多边缘云服务器之间的负载均衡问题,在满足任务执行时间和能耗的条件下设计使系统整体耗能最小的最优卸载策略。Jie 等<sup>[18]</sup>将移动边缘计算环境中用户之间的计算卸载策略抽象为符合主从关系的 Stackelberg 博弈,为优化任务的响应时间和资源利用率设计了一种任务调度算法。You 等<sup>[19]</sup>研究在同一网络中只有单一边缘云服务器时,多终端设备将部分任务卸载到边缘云中执行时所需的信道资源分配问题,优化任务执行所需时间。上述文献均假设边缘云服务器拥有无限的计算资源,而云机器人系统在现实的计算平台中运行,其次对选择部分卸载时结合能耗和时间延迟的计算卸载系统研究不足。

综上,为降低通信设备压力,提高云端服务调度灵活性,本文在云机器人系统中选择部分卸载的计算卸载策略。本文重点研究在同一边缘“网络”只有单个边缘云,且边缘云的计算资源是有限时,实现机器人计算任务部分卸载的策略研究。

## 2 计算卸载系统模型

云机器人系统中的计算卸载是典型的边缘计算(MEC)问题,根据上述对计算卸载的介绍,结合云机器人计算任务本身的特点可知:1)云机器人中的部分任务只能在本地执行,且存在云端资源竞争不足等情况,故采用动态部分卸载的策略;2)云机器人中的计算任务可被分割,部分在本地执行,同时其他部分被卸载至边缘服务器执行,根据计算卸载的影响因素决定单节点卸载还是多节点卸载。本文的博弈论算法是研究多个机器人竞争同一边缘云服务器资源,欲将计算任务卸载到边缘服务器中执行以减少自身设备的计算成本。任何一个机器人的卸载决策都会影响同一边缘“网络”中其他机器人终端的计算卸载决策。在云机器人系统的计算卸载模型中,处于同一边缘“网络”中的节点大多只有一个,符合多对一的模型。下面就基于 Stackelberg 博弈论的计算卸载策略方法进行论述。

### 2.1 系统网络模型

一个计算卸载网络由一个具备计算加速服务的边缘云和  $K$  个机器人终端集合  $N = \{1, 2, \dots, K\}$  组成,单个通信 AP 节点下管理至多  $K$  个移动机器人终端,其网络模型见图 2。

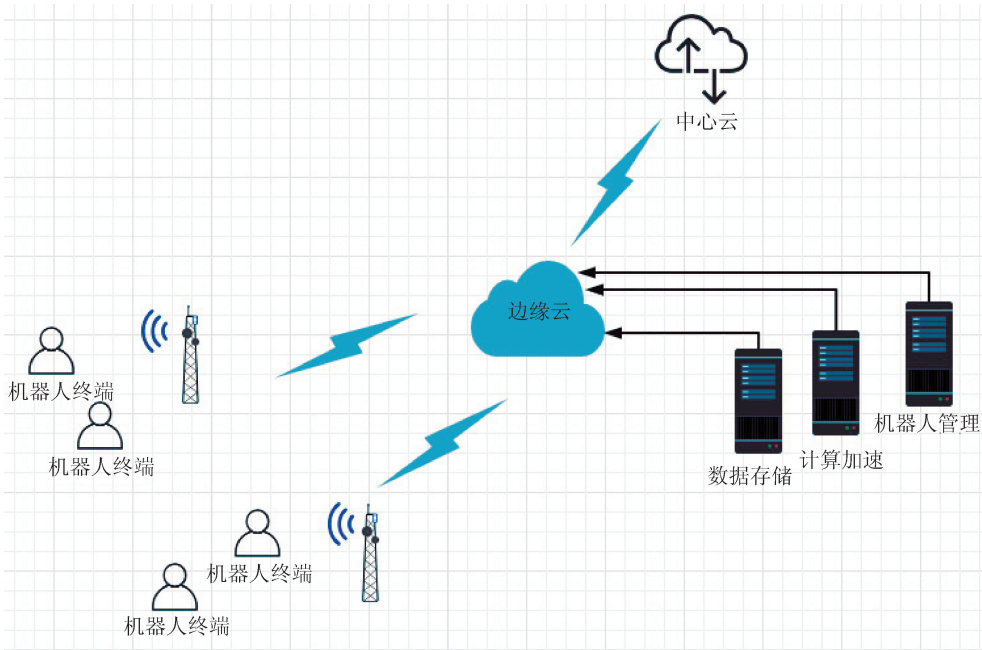


图 2 系统网络模型

Fig. 2 System network model

计算卸载方式为动态部分卸载,即机器人终端任务可按功能拆分为多个大小各异的简单任务,一部分在本地执行,另一部分在边缘服务器上执行。边缘云计算平台与通信 AP 节点之间采用有线网络,不考虑在有线传输过程中所产生的通信损耗和时间延迟,通信 AP 节点通过无线网络与机器人终端进行连接,该无线网络可分配的总带宽为  $B$ 。假设每个机器人终端与通信 AP 节点之间的信道相互独立,互不干扰,每个机器人可均匀分得的最大带宽为  $B/K$ ,且通信能力不随该节点管理机器人数量增加而下降。随着机器人终端的移动,其与通信 AP 节点之间的信道会随着位置发生改变,假设单次任务卸载周期内信道环境参数保持不变,但不同卸载周期信道环境参数可能改变。

## 2.2 任务执行模型

基于上述的网络模型,计算任务可在本地或边缘侧执行,首先对计算任务进行建模。云机器人系统中,功能各异的机器人终端选择适当的 CPU 计算能力,编号为  $k$  的机器人终端其计算能力用每秒 CPU 周期数  $F_k$  衡量,单位为 cycles/s,执行任务所需计算数据总量为  $R_k$ ,单位为 Byte,每执行 1bit 数据所需的 CPU 周期数为  $C_k$ ,单位为 cycle。计算任务卸载到边缘侧执行的比例为  $x_k$ ,  $0 \leq x_k \leq 1$ ,故执行卸载的数据量为  $x_k R_k$ ,本地执行的数据量为  $(1-x_k) R_k$ 。时间延迟和能量消耗是评价计算卸载策略优劣的重要指标,下文对任务本地执行和任务远端执行中的时间延迟和能量消耗计算方法进行建模。

### 1) 任务本地执行模型

任务在本地执行计算过程中所耗费的时间和能量即时间延迟和能耗。计算能力为  $F_k$  的机器人终端执行数据量大小为  $(1-x_k) R_k$  的计算任务所消耗的时间和能量分别为:

$$t_k^{\text{loc}} = (1-x_k) R_k C_k / F_k \quad (1)$$

$$e_k^{\text{loc}} = (1-x_k) R_k C_k V_k \quad (2)$$

式中:  $V_k$  表示每 CPU 周期消耗能量,单位 J。

### 2) 任务远端执行模型

任务在进行计算卸载后,其时间延迟  $t_k^{\text{off}}$  由四部分组成,分别是上载任务数据到边缘云的传输时间、在队列中等待执行的时间、边缘侧执行任务计算的时间以及反馈计算结果到机器人终端的传输时间。在上述过程中机器人终端的能量消耗  $e_k^{\text{off}}$  主要由任务数据上载和计算结果下载时通信设备耗能组成:

$$t_k^{\text{off}} = t_k^{\text{up}} + t_k^{\text{delay}} + t_k^{\text{comp}} + t_k^{\text{down}} \quad (3)$$

$$e_k^{\text{off}} = e_k^{\text{up}} + e_k^{\text{down}} \quad (4)$$

机器人终端与通信 AP 节点之间的信道带宽为  $B/K$ ,单次计算卸载周期内信道噪声功率不变,为  $\sigma$ ,其本地通信设备发送功率为  $P_k$ ,机器人终端与通信 AP 节点之间的信号增益为  $G_k$ ,根据 Shannon 公式可得:

$$r_k = \frac{B}{K} \log_2 \left( 1 + \frac{P_k G_k}{\sigma^2} \right) \quad (5)$$

因此机器人终端  $k$  上载任务数据到边缘云上的传输时间为:

$$t_k^{\text{up}} = x_k R_k / r_k \quad (6)$$

数据的上载和下载处于相同信道环境中,可得数据下载速率为:

$$r_{\text{AP},k} = \frac{B}{K} \log_2 \left( 1 + \frac{P_{\text{BS}} G_k}{\sigma^2} \right) \quad (7)$$

式中:  $P_{\text{BS}}$  为通信 AP 节点的通信设备发送功率。

计算结果的数据量相较于输入数据量明显减少,用  $\alpha$  表示输出数据量与输入数据量的比值,则反馈到机器人终端的数据量为  $\alpha x_k R_k$ ,因此计算结果反馈时间为:

$$t_k^{\text{down}} = \alpha x_k R_k / r_{\text{AP},k} \quad (8)$$

数据传输到边缘云之后会进入任务等待队列,在等待投入计算的过程中的时间用  $t_k^{\text{delay}}$  表示,规定任务的等待时间不超过  $T_d$ 。

每个机器人终端均匀分得边缘云的部分计算能力。边缘云分配的计算能力用  $F^{\text{edg}}$  表示,在边缘侧的任务计算时间  $t_k^{\text{comp}}$  为:

$$t_k^{\text{comp}} = x_k R_k C_k / F^{\text{edg}} \quad (9)$$

在此系统中任务可在本地和边缘侧同时进行计算,因此机器人  $k$  执行总数据量为  $R_k$  的任务需要的时间为:

$$t_k = \max \{ t_k^{\text{loc}}, t_k^{\text{off}} \} \quad (10)$$

在上述机器人  $k$  将任务数据上载到边缘云的过程中所消耗的能量为  $e_k^{\text{up}}$ :

$$e_k^{\text{up}} = TP_{s,k} t_k^{\text{up}} \quad (11)$$

式中:  $TP_{s,k}$  表示机器人  $k$  通信设备的发送功率。

计算结果反馈过程中所消耗能量为:

$$e_k^{\text{down}} = TP_{r,k} t_k^{\text{down}} \quad (12)$$

上式中  $TP_{r,k}$  表示机器人  $k$  通信设备的接收功率。则可得机器人终端执行计算卸载所消耗的能量为:

$$e_k^{\text{off}} = TP_{s,k} t_k^{\text{up}} + TP_{r,k} t_k^{\text{down}} \quad (13)$$

为对计算卸载过程中需付出的代价进行评估,设计了由多项式组成的评价代价函数公式:

$$\text{Cost}_k = \lambda^e e_k + \lambda^t t_k \quad (14)$$

能耗权重因子  $\lambda^e$  和时间延迟因子  $\lambda^t$  满足约束条件,  $\lambda^e + \lambda^t = 1$ 。则计算卸载过程中总的代价函

数表示为:

$$Cost_k = \lambda^i(t_k) + \lambda^e(e_k^{loc} + e_k^{off}) \quad (15)$$

### 2.3 云机器人系统中的 Stackelberg 博弈模型

部署在云机器人系统环境中的边缘云服务器计算资源通常是有限的,且同时服务的用户数也存在上限,因此边缘云与机器人终端之间,每个机器人终端之间都存在着明显的博弈过程,即边缘云通过对服务定价影响机器人终端的卸载决策,而每个机器人终端的卸载决策同时也会对其他机器人的决策产生影响。为激励机器人终端将计算任务卸载到边缘云上执行,保证云机器人系统高效运行的前提下,解决边缘云定价策略与机器人终端卸载策略的平衡问题,给出当前系统条件下的最优价格和计算卸载策略集。

机器人终端的卸载策略集  $X = \{x_1, x_2, \dots, x_K\}$  代表每个机器人卸载数据到边缘云中占总数据量的比例,  $0 < x_k < 1$  时表示卸载部分数据到边缘侧执行,  $x_k = 0$  表示任务在本地执行,  $x_k = 1$  表示将全部数据卸载到边缘侧执行,这两种视为计算卸载决策的特殊解。边缘云通过对计算资源定价,用  $P = \{p_1, p_2, \dots, p_K\}$  表示,控制机器人终端做出的计算卸载决策,因此机器人终端的卸载策略  $x_k$  是边缘云定价  $p_k$  的函数。可将云机器人系统中的博弈公式化为:  $G = \{P, X, U(P), U_k(x_k, p_k)\}$ , 其中  $U_k(x_k, p_k)$  代表机器人终端  $k$  的效用函数,  $U(P)$  代表边缘云的效用函数。

#### 1) 机器人效用函数

机器人终端执行计算卸载过程中的代价包括计算等待时间延迟、传输能耗以及使用边缘云服务支付的费用,可将机器人终端  $k$  的成本表示为:

$$Cost_k = \lambda^e((1-x_k)R_k V_k + TP_{s,k} t_k^{up} + TP_{r,k} t_k^{down}) + \lambda^i(t_k) + p_k x_k R_k C_k \quad (16)$$

机器人终端  $k$  的效用函数由任务执行所得收益与执行总代价之差组成,所得收益可用对数函数表示为  $Y_k(x_k)$ , 参数  $\beta$  与机器人终端任务相关,  $\beta > 0$ 。

$$Y_k(x_k) = \beta \log(1+x_k) \quad (17)$$

则机器人终端  $k$  的效用函数表示为:

$$U_k(x_k, p_k) = Y_k(x_k) - Cost_k \quad (18)$$

在给定的边缘云价格策略集下,机器人终端确定计算卸载策略集,最小化自身成本函数 P1:

$$\min(Cost_k) \quad \text{s. t. } 0 \leq x_k \leq 1 \quad (19)$$

同时期望获得最大的自身效用 P2:

$$\begin{cases} \max U_k(x_k, p_k) \\ \lambda^e + \lambda^i = 1, 0 < \lambda^i < 1, 0 < \lambda^e < 1 \\ \text{s. t. } \begin{cases} 0 \leq x_k \leq 1 \\ t_k \leq T_{\max} e_k \leq E_{\max} \end{cases} \end{cases} \quad (20)$$

式中:  $t_k$  代表机器人  $k$  执行卸载计算的总时延;  $e_k$  代表机器人  $k$  执行卸载计算的总能耗;  $T_{\max}$  代表任务的最大等待延迟;  $E_{\max}$  代表最大剩余可用能量。在众多参数的限制下需要最优化机器人终端自身效用。

#### 2) 边缘云效用函数

机器人使用计算资源支付的费用的总和即为边缘云效用函数的表达式:

$$U(P) = \sum_{k=1}^K p_k x_k R_k C_k \quad (21)$$

在满足边缘云最大计算资源的条件下,边缘云调整其定价策略,激励机器人终端卸载任务到边缘云中,最大化自身收益 P3:

$$\max U(P) \quad \text{s. t. } \sum_{k=1}^K x_k R_k C_k \leq F_{\text{edg\_all}} \quad (22)$$

式中:  $F_{\text{edg\_all}}$  为边缘云最大的计算资源数。

### 2.4 云机器人系统中 Stackelberg 博弈求解

逆向归纳法通过将多阶段动态博弈过程沿时间轴划分为多次单人博弈,确定每个参与者在不同时间点的最优选择,是分析和求解 Stackelberg 博弈均衡策略的常用方法。P3 问题中制定的价格会影响 P1 优化问题中的卸载策略,而同样 P1 问题的卸载策略会反作用于 P3 问题,问题 P1 和 P3 是耦合的。结合逆向归纳法,将机器人终端的卸载策略问题分解为给定边缘云定价策略集下的子问题,通过求解 P1 问题得到最优卸载策略集,而边缘云服务器可在已得最优卸载策略下解决 P3 问题中的最优价格。经过多次子问题的演化分析,得到全局最优的卸载策略集和价格策略集。

在任务模型小节中,任务执行的时间延迟  $t_k = \max\{t_k^{loc}, t_k^{off}\}$ , 二者均为机器人终端卸载策略  $x_k$  的函数,可令  $t_k^{loc} = t_k^{off}$ , 求得机器人终端卸载策略的中间值,可得:

$$x_k = \frac{C_k}{\theta_k F_k + C_k} = z_k \quad (23)$$

$$\theta_k = \frac{1}{r_k} + \frac{C_k}{F_e} + \frac{\alpha_k}{r_{AP,k}} \quad (24)$$

将解  $x_k$  代入机器人终端的成本函数后可得:

$$U_k(x_k, p_k) = \begin{cases} \lambda^e((1-x_k)R_k C_k V_k + TP_{s,k} t_k^{up} + TP_{r,k} t_k^{down}) + \\ \frac{\lambda^t(1-x_k)R_k C_k}{F_k} + p_k x_k R_k C_k, 0 \leq x_k \leq z_k \\ \lambda^e((1-x_k)R_k C_k V_k + TP_{s,k} t_k^{up} + TP_{r,k} t_k^{down}) + \\ p_k x_k R_k C_k + \lambda^t \theta_k x_k R_k, z_k \leq x_k \leq 1 \end{cases} \quad (25)$$

在给定价格策略集下,  $p_k$  视做常数,则上方成本函数可表示为变量  $x_k$  的分段函数:

$$U_k(x_k, p_k) = \begin{cases} (\lambda^e(TP_{s,k} t_k^{up} + TP_{r,k} t_k^{down} - R_k C_k V_k) + \frac{\lambda^t x_k R_k C_k}{F_k}) x_k + \\ \lambda^e R_k C_k V_k + p_k R_k C_k, 0 \leq x_k \leq z_k \\ (p_k R_k C_k + \lambda^t \theta_k R_k + \lambda^e(TP_{s,k} t_k^{up} + TP_{r,k} t_k^{down} - R_k C_k V_k)) x_k + \\ \lambda^e R_k C_k V_k, z_k \leq x_k \leq 1 \end{cases} \quad (26)$$

可将上述公式简化为:

$$U_k(x_k, p_k) = \begin{cases} A_{1k} x_k + b_1, 0 \leq x_k \leq z_k \\ A_{2k} x_k + b_2, z_k \leq x_k \leq 1 \end{cases} \quad (27)$$

分析可知,当  $A_{1k} < 0 < A_{2k}$  可得最优的计算卸载策略,即边缘云定价应在区间:

$$\lambda^e(V_k - \frac{TP_{s,k}}{r_k C_k} - \frac{TP_{r,k}}{r_{AP,k}}) - \lambda^t \frac{\theta_k}{C_k} < p_k < \frac{\lambda^t}{F_k} + \lambda^e(V_k - \frac{TP_{s,k}}{r_k C_k} - \frac{TP_{r,k}}{r_{AP,k}}) \quad (28)$$

令  $\omega_{1k} = \lambda^e(V_k - \frac{TP_{s,k}}{r_k C_k} - \frac{TP_{r,k}}{r_{AP,k}}) - \lambda^t \frac{\theta_k}{C_k}$  且  $\omega_{2k} = \frac{\lambda^t}{F_k} + \lambda^e(V_k - \frac{TP_{s,k}}{r_k C_k} - \frac{TP_{r,k}}{r_{AP,k}})$ , 则可知机器人终端做出决策为:

$$x_k = \begin{cases} 0 & p_k > \omega_{2k} \\ z_k & \omega_{1k} \leq p_k \leq \omega_{2k} \\ 1 & \omega_{1k} \geq p_k \end{cases} \quad (29)$$

当边缘云计算资源定价小于  $\omega_{1k}$  时,机器人终端将把所有计算任务卸载到边缘云中执行,而此时由于边缘云请求的计算服务量过大,导致其处理任务的效率变低,因此边缘云定价应避免设置在该范围。当边缘云计算资源定价大于  $\omega_{2k}$  时,机器人终端在进行计算卸载时的成本开销过大,导致所有任务都在本地进行,而边缘云的收益为0,因此边缘云定价应避免此区间。当  $\omega_{1k} \leq p_k \leq \omega_{2k}$  时,机器人终端将部分任务卸载到边缘侧执行,保证了边缘云的收益和任务处理效率,减少机器人处理任务的能耗。

### 3 系统实现与测试

基于给定最优卸载策略集下的边缘云定价策略分析可知,对于机器人终端  $k$ ,其最优卸载策略为  $x_k^* = z_k$ ,边缘云制定的最优计算资源定价为  $p_k^* = \omega_{2k}$ 。边缘云服务器的计算资源通常是有限的,并非每个机器人终端都可将任务卸载到边缘云中,因此需要为机器人终端设计在当前系统状态下判断是否进行卸载的算法。动态规划算法是将一个复杂问题转化

为一系列简单问题的最佳途径,结合本文中的系统模型,提出一种基于 Hamming 距离的动态规划算法,在机器人终端数为  $N$  的云机器人计算卸载网络中,使用  $N \times N$  的表格存储是否卸载的二进制字符串。首先,生成一个随机的大小为  $N$  的二进制字符串,其作为第一个卸载方案被填充到动态规划表格中,规则为如下两条。

1) 二进制字符串的第一个位置若为1,则起始位置为(1,2),否则起始位置为(2,1)。进入子串的下一个位置。

2) 当下一位置上的二进制值为1时,则1被填充到水平方向上的表格中,若下一位置的二进制值为0,则0被填充到垂直方向上的表格中。

例如,在一个机器人终端数为10的云机器人系统环境中,将动态规划填充表使用图3中的一个二维  $10 \times 10$  的表格表示。

		1	1	1					
0				0					
0	1	1	0	1	1				
		0	1	1	0				
				0	0	1			
				0					
				0					

(a) 示例一

		1	1	1					
0		0	0						
0	1	1/0	0	1	1				
		0	1	1	0/1	1			
				0	0	1/0			
				0					
				0					

(b) 示例二

图3 算法示例

Fig. 3 Algorithm example

假设第一个随机字串为“1110011001”(见图 3(a)中黑色字串)或者“0011011000”(见图 3(a)中红色字串)。第二个随机字串为“1100011110”(见图 3(b)绿色字串)。因为第二个字串的第一位为 1,所以起始位置为(1,2)。基于该方法处理,可有效减少公共子串的重复计算。

每生成一个随机的二进制字串,需要计算该字串总的执行时间和能量消耗,同时还需要计算表格中每个方格的总能量消耗和执行时间。但是如果生成的随机字串与在表格中的字串有相同的部分,则在计算二进制字串的时间和能耗时只计算到第一个相同数值的方格,然后将新计算的总能量和记录在该方格内的总能量相比较。如果新计算的总能量小于记录在方格内的能量,则使用新的子串代替旧的子串,并使用新计算的总能量、能量消耗以及执行时间替换掉已记录的值。基于该规则更新剩下的所有子串的能量和执行时间。否则,如果新计算的总能量大于记录在方格中的总能量,则保留在该方格中已记录的子串。当算法中的字串 Hamming 距离大于给定阈值,便终止循环并返回计算结果。

机器人终端的 CPU 处理能力随机分布于集合  $\{0.1 \text{ GHz}, 0.2 \text{ GHz}, \dots, 1.0 \text{ GHz}\}$  之中,每执行 1 字节数据需要消耗的 CPU 周期数在  $500 \sim 1\,500$  cycles/bit 随机分布,机器人终端  $k$  的输入待处理数据总量  $R_k$  设置为  $100 \sim 300$  KB 的随机值。机器人终端通信设备的发射功率为  $0.1 \text{ W}$ ,接收功率为  $0.06 \text{ W}$ ,通信 AP 节点的发射功率为  $1 \text{ W}$ 。边缘云计算能力  $100 \text{ GHz}$ 。机器人数量最小为 15,以 15 的倍数增长。时间权重因子  $\lambda^t$  随机分布于区间  $[0, 1]$ ,能量权重因子  $\lambda^e = 1 - \lambda^t$ 。

在将系统部分参数去除随机性后,机器人终端成本开销和效用随计算卸载比例的变化曲线图见图 4 和图 5。

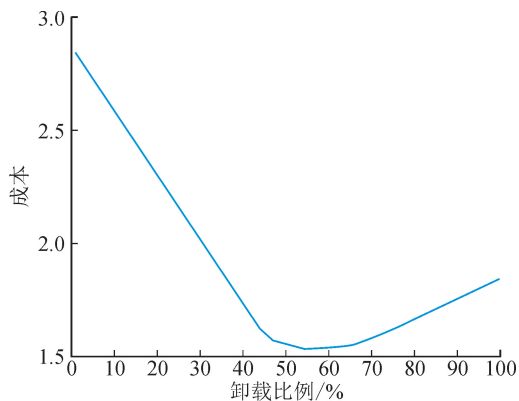


图 4 计算卸载成本均值随卸载比例实验曲线  
Fig. 4 Experiment curve of average offloading cost at offloading ratio

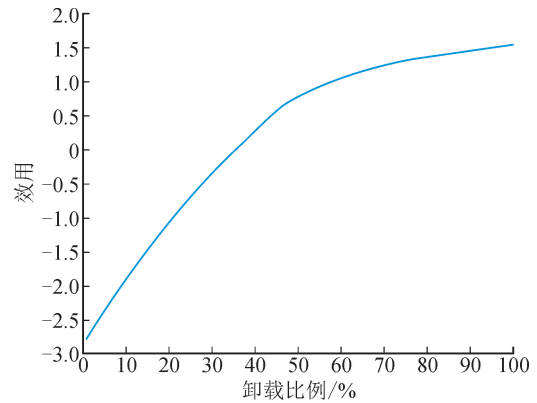


图 5 机器人终端效用均值随卸载比例实验曲线  
Fig. 5 Experiment curve of average utility of robot terminal at offloading ratio

随着计算卸载比例增大,更多的计算任务卸载到边缘测执行,使机器人终端计算成本降低。但由于越来越多的机器人对任务执行卸载,使边缘云资源竞争愈发激烈,导致计算资源价格上升,使机器人终端的成本函数增大,因此机器人终端成本开销的平均值会随着卸载比例的增大先减小再增大。而用户效用的平均值会随着卸载比例增大,付出更多的成本,而执行计算卸载的收益趋于平衡。此实验验证了系统模型的正确性。

图 6 表示当边缘云网络内的机器人终端数量增加时,任务全部本地执行和使用本文提出的 Stackelberg 博弈方法在机器人终端成本开销上的对比结果。结合文中的分析可知,在使用 Stackelberg 博弈方法时,机器人终端成本开销的平均值明显低于本地执行的开销,因为虽然本机计算无需向边缘云支付计算资源费用,但本地计算时间延迟较高,使成本开销增大。

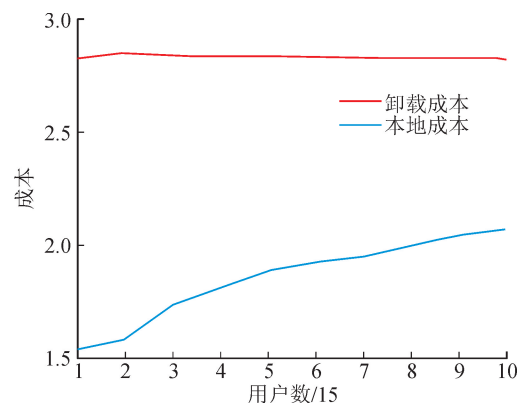


图 6 机器人终端本地及卸载计算成本随用户数实验曲线  
Fig. 6 Experiment curve of tasks' cost running in local and offloading follow the growing users

## 4 结 语

本文基于 Stackelberg 博弈思想设计了云机器

人系统环境下机器人终端执行计算任务卸载的优化算法,解决在多机器人终端竞争同一边缘云服务器计算资源场景下的边缘云效用及机器人终端效用均衡问题。并就移动终端的卸载决策问题提出了一种基于动态规划的优化算法,该算法通过开辟存储计算过程的数组,减少了对公共子串的重复计算。通过对不同计算能力和能量状态的机器人终端设置不同的权重因子,以及对计算卸载决策博弈过程的分析,分别选择对机器人终端和边缘云最优的卸载策略和定价策略。结合实验分析,该系统有效降低机器人终端计算能量消耗以及提高边缘云自身效用,并验证了系统模型的正确性。

#### 参考文献:

- [1] 郭宇. 云机器人系统网络优化与计算卸载相关技术研究[D]. 北京:北京科技大学,2020.  
GUO Yu. Research on network optimization and computing offload of cloud robot system[D]. Beijing: University of Science and Technology Beijing, 2020
- [2] 王大伟,王卓,王鹏,等. 基于边缘计算的云原生机器人系统[J]. 智能科学与技术学报,2020,2(3):275-283.  
WANG Dawei, WANG Zhuo, WANG Peng, et al. Cloud native robot system based on edge computing[J]. Journal of Intelligent Science and Technology, 2020, 2(3):275-283.
- [3] 张松. 基于云计算的服务机器人系统设计与实现[D]. 西安:西安科技大学,2020.  
ZHANG Song. Design and implementation of service robot system based on Cloud Computing[D]. Xi'an: Xi'an University of Science and Technology, 2020.
- [4] XU Fei, YANG Weixia, LI He. Computation offloading algorithm for cloud robot based on improved game theory [J]. Computers & Electrical Engineering, 87: 106764.
- [5] 尤斐然. 基于博弈论的异构网络中边缘计算与缓存技术研究[D]. 南京:南京理工大学,2019.  
YOU Feiran. Research on edge computing and caching technology in heterogeneous network based on Game Theory[D]. Nanjing: Nanjing University of Technology,2019
- [6] 武星,王旻超,张武,等. 云计算研究综述[J]. 科技创新与生产力,2011(6):7.  
WU Xing, WANG Yanchao, ZHANG Wu, et al. Overview of cloud computing research[J]. Scientific and Technological Innovation and Productivity, 2011(6):7.
- [7] KUFFNER J. Cloud-enabled humanoid robots[C]// Humanoid Robots, 2010.
- [8] 张恒,刘艳丽,刘大勇. 云机器人的研究进展[J]. 计算机应用研究,2014,31(9):2567-2575.  
ZHANG Hen, LIU Yanli, LIU Dayong. Research progress of cloud robotics[J]. Computer Application Research, 2014, 31(9):2567-2575.
- [9] ARUMUGAM R, ENTI V R, LIU B, et al. DAViNCi: A cloud computing framework for service robots[C]// IEEE International Conference on Robotics & Automation. IEEE, 2010.
- [10] QUINTAS J M, MENEZES P J, DIAS J M. Cloud robotics: toward context aware robotic networks[C]// Proceedings of the IASTED International Conference Robotics (Robo2011), 2011.
- [11] DU Zhihui, YANG Weiqiang, CHEN Yinong, et al. Design of a Robot Cloud Center [C]//International Symposium on Autonomous Decentralized Systems. IEEE Computer Society, 2011.
- [12] FLORES H, HUI P, TARKOMA S, et al. Mobile code offloading: from concept to practice and beyond [J]. IEEE Communications Magazine, 2015, 53(3): 80-88.
- [13] 董思岐,李海龙,屈毓铨,等. 移动边缘计算中的计算卸载策略研究综述[J]. 计算机科学,2019,46(11): 32-40.  
DONG Siqi, LI Hailong, QU Yuben, et al. Survey of research on computation offloading strategy in mobile edge computing[J]. Computer Science,2019,46(11): 32-40
- [14] CHEN Xu, JIAO Lei, LI Wenzhong, et al. Efficient multi-user computation offloading for mobile-edge cloud computing[J]. IEEE/ACM Transactions on Networking,2016,24(5):2795-2808.
- [15] MA Xiao, LIN Chuang, XIANG Xudong, et al. Game-theoretic analysis of computation offloading for cloudlet-based mobile cloud computing[C]// MSWiM'15: Proceedings of the 18th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, 2015: 271-278.
- [16] CHEN Mengshi, DONG Min, LAING Ben. Multi-user mobile cloud offloading game with computing access point[C]//2016 5th IEEE International Conference on Cloud Networking (Cloudnet), 2016: 64-69.
- [17] DAI Yueyue, XU Du, MAHARJAN S, et al. Joint computation offloading and user association in multi-task mobile edge computing[J]. IEEE Transactions on Vehicular Technology, 2018, 67:12313-12325.
- [18] JIE Yingmo, TANG Xinyu, CHOO K K R, et al. Online task scheduling for edge computing based on repeated Stackelberg game[J]. Journal of Parallel and Distributed Computing, 2018, 122: 159-172.
- [19] YOU Changsheng, HUANG Kaibin. Multiuser resource allocation for mobile-edge computation offloading[C]//2016 IEEE Global Communications Conference (GLOBECOM), 2016: 1-6.

(责任编辑 王绪迪)