

5-22-2023

Partial Task Offloading Strategy of Cloud Robots Based on Game Theory under Cloud-Edge Coordination

Chunmao Jiang

*1.College of Computer Science and Mathematics, Fujian University of Technology, Fuzhou 350118, China;;
hsdrose@126.com*

Zhenxing Yang

*2.School of Computer Science and Information Engineering, Harbin Normal University, Harbin 150500,
China;*

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the Artificial Intelligence and Robotics Commons, Computer Engineering Commons, Numerical Analysis and Scientific Computing Commons, Operations Research, Systems Engineering and Industrial Engineering Commons, and the Systems Science Commons

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

Partial Task Offloading Strategy of Cloud Robots Based on Game Theory under Cloud-Edge Coordination

Abstract

Abstract: How to rationally utilize the resources of central and edge clouds to reduce energy consumption of system equipment and shorten average task completion time is a fundamental challenge for computational task offloading of cloud robots. *In this paper, we transform the computational task offloading problem of multiple cloud robots into a multi-actor game model by using the computational task completion time and energy consumption of cloud robots as cost measurement indicators and setting different cost weights according to actual needs. We also develop a game theory-based partial task offloading algorithm (GT-PTO). With the Nash equilibrium state under this algorithm, the optimal offloading threshold for the participants can be found, and the total system cost can be optimized.* Simulation results show that the proposed algorithm can be used for task offloading, so as to reduce the energy consumption of computational tasks of cloud robots, shorten the average task completion time, and significantly improve the quality of cloud-edge collaboration services.

Keywords

game theory, cloud robot, task offloading, cloud-edge coordination, energy consumption optimization

Recommended Citation

Chunmao Jiang, Zhenxing Yang. Partial Task Offloading Strategy of Cloud Robots Based on Game Theory under Cloud-Edge Coordination[J]. Journal of System Simulation, 2023, 35(5): 987-997.

云边协同下基于博弈论的云机器人部分任务卸载策略

姜春茂¹, 杨振兴²

(1. 福建工程学院 计算机科学与数学学院, 福建 福州 350118; 2. 哈尔滨师范大学 计算机科学与信息工程学院, 黑龙江 哈尔滨 150500)

摘要: 如何合理地利用中心云、边缘云的资源, 既降低系统设备能耗, 又能缩短任务平均完成时间, 是云机器人计算任务卸载面临的重大挑战。将云机器人的计算任务完成时间与能耗作为代价衡量指标, 根据自身需求设置不同的代价权重, 将多个云机器人的计算任务卸载问题转换成了一种多个玩家参与的博弈模型, 设计了一种基于博弈论的部分任务卸载算法(game theory-partial task offloading, GT-PTO)。通过算法下的纳什平衡状态, 找到参与者的最佳卸载阈值, 从而达到系统总代价的优化。仿真结果表明, 采用所提算法进行任务卸载, 能够减少云机器人计算任务的能耗, 缩短平均任务完成时间, 大大提高云边协同服务质量。

关键词: 博弈论; 云机器人; 任务卸载; 云边协同; 能耗优化

中图分类号: TP316.4; TP391.9 文献标志码: A 文章编号: 1004-731X(2023)05-0987-11

DOI: 10.16182/j.issn1004731x.joss.22-0014

引用格式: 姜春茂, 杨振兴. 云边协同下基于博弈论的云机器人部分任务卸载策略[J]. 系统仿真学报, 2023, 35(5): 987-997.

Reference format: Jiang Chunmao, Yang Zhenxing. Partial Task Offloading Strategy of Cloud Robots Based on Game Theory under Cloud-Edge Coordination[J]. Journal of System Simulation, 2023, 35(5): 987-997.

Partial Task Offloading Strategy of Cloud Robots Based on Game Theory under Cloud-Edge Coordination

Jiang Chunmao¹, Yang Zhenxing²

(1. College of Computer Science and Mathematics, Fujian University of Technology, Fuzhou 350118, China;

2. School of Computer Science and Information Engineering, Harbin Normal University, Harbin 150500, China)

Abstract: How to rationally utilize the resources of central and edge clouds to reduce energy consumption of system equipment and shorten average task completion time is a fundamental challenge for computational task offloading of cloud robots. In this paper, we transform the computational task offloading problem of multiple cloud robots into a multi-actor game model by using the computational task completion time and energy consumption of cloud robots as cost measurement indicators and setting different cost weights according to actual needs. We also develop a game theory-based partial task offloading algorithm (GT-PTO). With the Nash equilibrium state under this algorithm, the optimal offloading threshold for the participants can be found, and the total system cost can be optimized. Simulation results show that the proposed algorithm can be used for task offloading, so as to reduce the energy consumption of computational tasks of cloud robots, shorten the average task completion time, and significantly improve the quality of cloud-edge collaboration services.

Keywords: game theory; cloud robot; task offloading; cloud-edge coordination; energy consumption optimization

收稿日期: 2022-01-08 修回日期: 2022-06-30

基金项目: 黑龙江省自然科学基金(LH2020F031)

第一作者: 姜春茂(1972-), 男, 教授, 博士, 研究方向为云计算、智能决策等。E-mail: hsdrose@126.com

0 引言

随着无线网络、大规模存储和通信技术的迅猛发展以及互联网资源的无所不在,一种基于云计算和服务机器人进行广泛合作与交互的技术应运而生。在 Humanoids 2010 会议上,卡耐基梅隆大学 James Kuffner 教授提出了“云机器人”的概念,引起了广泛的讨论^[1]。一方面,云计算可以通过大规模并行计算与分布式存储为机器人提供更强大的计算能力,从而增强机器人的处理能力;另一方面,机器人在云计算等的支持下可以完成一些人类无法完成或者相对危险的任务。例如:Facebook、Google 等公司正计划将无人机群作为低空平台集成到蜂窝网络中,以提供宽带连接并补偿小区过载或站点中断。再如,机器人被用来清除核反应堆凸出物内部和外部的碎片,并检测福岛第一核电站的放射性同位素水平。目前,云机器人技术已经被广泛应用于感知和计算视觉、导航、抓取和操纵、制造或服务机器人、社会和医疗应用等诸多领域^[2]。

决定云机器人系统性能的关键因素之一是决定将任务上传到云端还是使用本地资源进行处理,尤其是当机器人面临诸如人脸识别、增强现实、自然语言处理等复杂计算的密集型任务的时候。一些基于中心云和机器人之间进行协同卸载的方法被提出来用于解决特定目标的问题,如拍卖机制^[3],信念传播网络^[4],序列博弈^[5]等。这些工作有的通过拍卖机制中的出价来揭示偏好,有的充分利用了可用信息(如意愿支付和相应时间)等。然而,移动数据的爆炸性增长给云和无线链路带来了沉重的负担,基于中心云和端侧机器人的卸载策略毫无疑问的会面临网络传输的极大不确定性,因而存在服务时间大量被延误的可能。

为了弥补云计算能力的不足,移动边缘计算(mobile edge computing, MEC)^[6]的范式被提出。移动边缘计算的提出给基于云端协同卸载策略提供了进一步的解决思路。边缘节点的性能要远高于

于中心云节点,网络传输速度也相对的快。机器人可以通过无线网络选择将计算任务卸载到云服务器或者边缘服务器中,从而大幅度减少任务执行时间,有效延长机器人的使用寿命,减少任务成本。综上所述,当多个云机器人任务通过同一信道卸载到边缘云或者中心云时会导致传输信道的堵塞,影响任务执行时间,产生额外的运行成本^[7-8]。因此如何做出多个云机器人任务的卸载决策,合理的利用中心云(central cloud computing, CCC)和 MEC 服务器的计算资源,使云机器人的任务完成时间及能耗的联合代价最小化是本文要解决的一项重要任务。

本文的主要贡献为:以计算任务的时延和能耗为目标函数,基于可划分的任务特征,提出一种基于博弈论的云-边-端协同卸载的策略,通过纳什均衡来确定最佳卸载阈值,从而决定任务在云、边、端哪个地方执行。

1 相关工作

对于计算任务卸载问题,文献[9]提出了一种部分任务卸载策略。这项工作研究了多用户 MEC 系统,设计了一种最佳策略来控制卸载数据量和时间/子信道分配,以最大程度地降低系统能耗。文献[10]提出了一种两层计算卸载框架来解决卸载问题。他们通过共同优化用户关联,计算资源分配,传输功率分配和计算分流任务来优化多个 MEC 服务器之间的负载平衡,并最大程度地降低系统能耗。这项工作一个局限性是他们忽略了云计算服务器能够带来的好处。文献[11]提出了一种针对高实时性任务的启发式协同计算卸载算法(heuristic collaborative computing offload algorithm, HCCOA),该算法基于中心云和边缘云协同计算卸载,以最小化任务总处理时间为目标。该研究比较了不同的卸载策略,从而选择时间最短的卸载策略。在这项工作中,他们还提出了协同计算卸载遗传算法(genetic algorithm for

collaborative computing offloading, GA-CCO), 以获得更好的任务卸载策略, 局限性是未能综合考虑能耗的影响。文献[12]基于光纤无线混合网络的一般结构, 研究了该网络结构下基于MEC和CCC的协同计算任务卸载问题, 并将其定义为以移动设备最小总能耗为目标的约束优化问题。这个工作提出了近似协同计算卸载 (approximate collaborative computing offload, ACCO) 方案和分布式计算卸载算法 (distributed computing offload algorithm, DCOA), 局限性是忽略了任务执行时间带来的影响。

近年来, 利用博弈论和改进博弈论算法来解决云机器人任务卸载决策问题的方法受到了越来越多的关注。博弈论是有效解决多个相互影响的参与者的决策问题的一种方法。为探寻博弈论在计算任务卸载方面的理论和方法, 文献[13]采用博弈论来表述计算卸载的决策过程, 讨论了多用户在多无线信道环境中访问边缘计算(MEC)的计算卸载问题, 将多用户分布式计算卸载决策过程建模为多用户计算卸载博弈, 设计了一种分布式计算卸载算法。该工作定义的效用函数为时间开销和能量开销的加权和, 通过最小化效用函数, 实现了时间开销和能量开销之间的平衡, 但设计的卸载方案并没有云服务器的直接参与。文献[14]在移动用户和边缘服务器之间建立了可重复的基于Stackelberg博弈的边缘计算任务调度模型, 并采用了历史数据集。利用需求预测方法解决了任务调度的Stackelberg平衡问题。该研究中定义的任务卸载是二进制的, 对于部分卸载来说任务是不可分割的。文献[15]研究了在5G无线系统中多无线信道移动边缘计算环境、多用户场景下的计算卸载问题。该研究以能量开销和时间开销的加权和最小为目标, 设计了时间复杂度较低的部分计算卸载算法。该研究以边缘服务器为卸载的目标节点, 没有考虑利用云服务器作为卸载节点之一可能带来的益处。

根据上述研究可知, 目前大部分基于博弈论

的云机器人任务卸载相关研究中存在如下的一些局限性: ①大部分的研究未考虑到计算任务的可分割性, 任务卸载大都是二进制, 未能考虑到计算任务细粒度划分后部分卸载所能带来的成本优化。②大多数研究只单纯考虑了MEC服务器的计算资源而忽略了云中心服务器强大的计算能力和利用这些资源所能带来的效益。MEC在处理云机器人卸载的大量计算密集型任务时, 其资源瓶颈愈发明显。当大量数据堆积到MEC同一信道时可能会经历较长的等待时间, 增加任务处理延迟, 影响机器人本身性能。

基于上述问题, 本文设计了云边端协作工作的基于博弈论的多机器人部分任务卸载策略, 模拟出当多个云机器人的计算任务在通过同一信道进行任务卸载时, 如何地充分利用云边协同系统的计算资源, 如何确定每个机器人的最佳任务卸载位置以及最优部分任务卸载量。本工作运用博弈论的方法, 将计算任务的时延和能耗设置为代价函数, 当卸载决策更新后, 信道传输速率受到影响, 机器人的任务卸载量发生变化, 影响总代价。具体为: ①将云机器人的任务进行细粒度划分, 通过一对阈值将机器人任务划分为三部分。②阈值划分的三部分计算任务可以选择本地计算、也可以选择卸载到边缘云或中心云计算, 即云一边一端的协同计算方式。③通过设计的博弈模型, 以博弈的方法找到云机器人系统任务卸载策略的Nash平衡状态, 即每个机器人任务划分的最佳卸载阈值策略。

2 系统模型与代价函数

2.1 系统模型

图1为本文使用的云边协作系统模型^[16], 包括1个云中心服务器、1个MEC服务器和 n 个云机器人。其中MEC服务器配有一个基站(base station, BS)。假设当前有任务的云机器人集合表示为 $R = \{1, 2, \dots, i, \dots, n\}$, 云机器人 $i \in R$ 。每个云机器人

都有一个计算任务用 τ_i 来表示,参考文献[17]提供的方法,使用二元组 (C_i, D_i) 来量化计算任务 τ_i ,其中 C_i 表示计算任务的计算量, D_i 表示计算任务的数据量。

所有云机器人 $i \in R$,当前的计算任务都可由一对阈值 $(\alpha_i^k, \beta_i^k) \in Q_i$ 划分为三部分, Q_i 为可划分的阈值策略空间,其策略空间与阈值划分精度有关:

$$Q_i = \{(\alpha_i^k, \beta_i^k) | 0 \leq \alpha_i^k \leq \beta_i^k \leq 1, \alpha_i^k = N \cdot \zeta, \beta_i^k = N' \cdot \zeta, i \in R, k \in K\}$$

为保证策略空间是有限策略集合,设 N 和 N' 为自然数, K 为策略空间数量的集合, ζ 为阈值划分步长,如 $\zeta=0.05$,则阈值划分步长为5%。当给定 ζ 后, K 随即确定,显然阈值划分的步长越小,阈值划分的精度就越高,其策略空间越大,云机器人可选择的卸载策略越多,但存在的卸载过程越复杂。

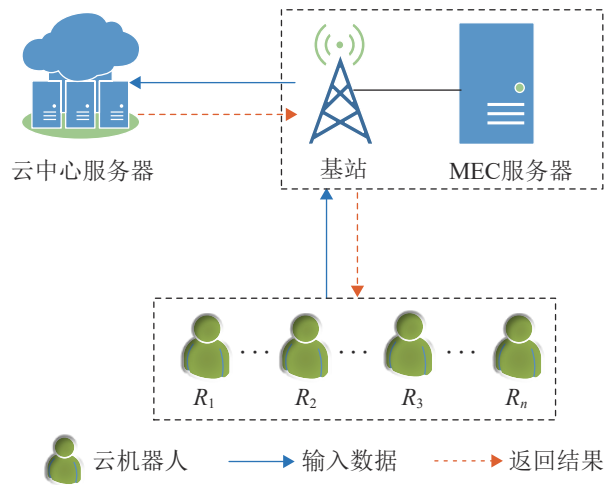


图1 系统模型
Fig. 1 System model

设机器人 i 的卸载策略为 q_i^k ,则 $q_i^k \in Q_i$,云机器人根据卸载策略将三部分计算任务分别在云一边一端进行处理,其中 $(0, \alpha_i^k)$ 表示本地处理的部分任务, (α_i^k, β_i^k) 表示卸载到MEC服务器执行的部分任务, $(\beta_i^k, 1)$ 表示云中心服务器(CCC)执行的部分任务。特别地当 $\alpha_i^k = \beta_i^k = 0$ 时,此时代表机

器人 i 的计算任务全部卸载到云服务器执行;当 $\alpha_i^k = 0, \beta_i^k = 1$ 时,代表机器人 i 当前的计算任务全部卸载到MEC服务器执行;当 $\alpha_i^k = \beta_i^k = 1$ 时,代表机器人 i 当前的计算任务全部本地执行。则当前所有云机器人的划分阈值策略构成策略集合 $S = \{q_i^k, i \in R\}$ 。

2.2 不同卸载策略下的代价函数

2.2.1 本地处理模型

当云机器人 i 在做出卸载决策 q_i^k 后其计算任务 τ_i 的全部或部分任务在本地执行,计算任务产生的代价主要包括任务完成时间与本地能耗。本地计算时间主要取决于其自身计算能力。令 F_i^{loc} 表示云机器人的CPU计算能力(周期数/s为单位)^[18],则本地计算时间为

$$T_i^{\text{loc}} = \frac{C_i \alpha_i^k}{F_i^{\text{loc}}} \quad (1)$$

本地计算能耗使用常用的能耗计算模型^[18] $\varepsilon = \eta f^2$,其中: η 为能耗系数,即每个CPU消耗的能量; f 为CPU频率。由此可知,云机器人本地执行任务能耗为

$$E_i^{\text{loc}} = C_i \eta (F_i^{\text{loc}})^2 \alpha_i^k \quad (2)$$

云机器人 i 在能耗和时间成本不同权重下的本地代价函数为

$$\text{Cost}_i^{\text{loc}}(q_i^k) = \lambda_i^e E_i^{\text{loc}} + \lambda_i^t T_i^{\text{loc}} \quad (3)$$

式中: $0 \leq \lambda_i^e \leq 1, 0 \leq \lambda_i^t \leq 1, \lambda_i^e + \lambda_i^t = 1$ 。

式(2)和(3)对执行时间和能耗进行归一化, λ_i^e 和 λ_i^t 分别为云机器人 i 对能耗和时间的权重参数。显然,不同的云机器人会对时间和能耗有不同的权重参数。例如,当云机器人当前电量较低时,则它会更注重能耗带来的影响,此时它的代价函数中能耗权重可以设置的较高,甚至设置 $\lambda_i^e = 1$ 。反之,当它在计算一个延迟敏感型任务时,它对任务完成时间要求会很高,因此可以设置较高的 λ_i^t ,甚至在极端情况设置为 $\lambda_i^t = 1$ 。此时意味着代价函数不会考虑能耗产生的影响。为了方便,假

设每个云机器人在完成当前计算任务 τ_i 的整个过程中对时间和能耗的权重参数不变。

2.2.2 MEC 服务器处理模型

云机器人 i 将当前决策 q_i^k 下的计算任务 τ_i 卸载到边缘服务器(MEC 端)执行时, 能耗会在传输过程产生, 而时延包括上行链路的传输时间以及在 MEC 服务器端执行任务的时间。由于其输出的结果比输入小得多, 因此, 返回到本地的时间可以忽略不计^[15,18-19]。信道数据传输采用非正交多址(non-orthogonal multiple access, NOMA)技术, NOMA 技术允许多个云机器人在同一信道下进行任务数据传输。根据香农定律, 当多个云机器人的任务经过同一信道进行传输时, 上行传输速率 r_i ^[20] 表示为

$$r_i = B \cdot \log\left(1 + \frac{p_i h_i}{I_i + \omega_0}\right) \quad (4)$$

式中: B 为传输信道 BS 的带宽; p_i 为云机器人的上行传输功率; h_i 为云机器人与信道 BS 之间的增益; 根据它们之间的距离定义^[21] $h_i = d_i^{-\sigma}$, d_i 为云机器人 i 到边缘云之间的距离; σ 为衰减系数; I_i 为信道内其它机器人传输数据时对机器人 i 所造成的干扰; ω_0 为信道内部的高斯噪声功率。任务卸载到 MEC 服务器执行时传输过程产生的能耗 E_i^{mec} , 上行传输时间 T_i^{upm} 以及 T_i^{mec} 与任务卸载量有关, 因此, 与任务卸载决策 q_i^k 相关。

$$E_i^{\text{mec}} = \frac{p_i D_i (\beta_i^k - \alpha_i^k)}{r_i} \quad (5)$$

云机器人卸载到 MEC 服务器时的上行传输时间为

$$T_i^{\text{upm}} = \frac{D_i (\beta_i^k - \alpha_i^k)}{r_i} \quad (6)$$

云机器人任务在 MEC 服务器的执行时间为

$$T_i^{\text{mec}} = \frac{C_i (\beta_i^k - \alpha_i^k)}{F_{\text{mec}}} \quad (7)$$

式中: F_{mec} 为 MEC 服务器所能提供的计算资源。综上, 云机器人 i 在决策 q_i^k 下 MEC 服务器端执行任务所产生的代价函数为

$$\text{Cost}_i^{\text{mec}}(q_i^k) = \lambda_i^e E_i^{\text{mec}} + \lambda_i^t (T_i^{\text{upm}} + T_i^{\text{mec}}) \quad (8)$$

2.2.3 云服务器处理模型

如果云机器人将计算任务卸载到中心云, 则必须要同时考虑将输入数据从 BS 发送到云端的上行传输时间和将输出结果从云服务器返回到 BS 的下行传输时间^[16]。无论任务卸载到 MEC 服务器还是云服务器, 都要包括从本地到 BS 的上行传输时间。因此, 如果云服务器所能提供的计算资源为 F_c , 则在决策 q_i^k 下云服务器执行任务时间为

$$T_i^c = \frac{C_i (1 - \beta_i^k)}{F_c} \quad (9)$$

云机器人将任务卸载到云服务器时从本地到 BS 的上行传输时间为

$$T_i^{\text{upc}} = \frac{D_i (1 - \beta_i^k)}{r_i} \quad (10)$$

云机器人任务卸载时在 BS 与云服务器传输链路中产生的时延为

$$T_i^{\text{bc}} = (D_i + D_i^r) (1 - \beta_i^k) v \quad (11)$$

式中: D_i^r 为返回计算结果的数据量; v 为单位数据传输时延。卸载任务能耗与 MEC 服务器处理模型相似, 云服务器处理任务的能耗为

$$E_i^c = p_i T_i^{\text{upc}} = \frac{p_i D_i (1 - \beta_i^k)}{r_i} \quad (12)$$

则在决策 q_i^k 下云服务器处理任务代价函数为

$$\text{Cost}_i^c(q_i^k) = \lambda_i^e E_i^c + \lambda_i^t (T_i^{\text{upc}} + T_i^c + T_i^{\text{bc}}) \quad (13)$$

云机器人 i 的信道传输速率 R_i 会受其它云机器人任务卸载策略的影响。计算任务在卸载到 MEC 或云服务器时都需要考虑计算任务从本地通过信道传输到 BS 这一段链路的传输速率。对于云机器人 i 来说, 其它云机器人 j 在进行任务卸载时对其造成的干扰为^[15]

$$I_i = \sum_{\{j \in R | q_j^k \in q_j^{\text{off}}\}} p_j h_j \quad (14)$$

其中, q_j^{off} 表示非全部任务本地执行的决策集合, 即有任务卸载的决策集合。将其带入式(4)中, 则云机器人 i 的上行传输速率为

$$r_i = B \cdot \ln \left(1 + \frac{p_i h_i}{\sum_{\{j \in R | q_j^k \in q_j^{\text{off}}\}} p_j h_j + \omega_0} \right) \quad (15)$$

3 基于博弈论的部分任务卸载策略

本文分析了基于博弈论的多个云机器人在准静态场景下(在一段时间内云机器人对时间能耗的代价权重不变, 即 λ_i^e 和 λ_i^t 不变)计算任务进行部分卸载的问题。在多个云机器人的计算任务竞争云、边服务器资源时, 如何做出任务的部分卸载决策, 进而达到各自利益的最大化, 即自身设备完成计算任务代价的最小化是本文要解决的关键问题。云机器人各自的卸载策略都会影响其它机器人的决定, 它们之间存在竞争博弈的关系。运用博弈论的方法, 找到它们卸载策略的纳什平衡(nash equilibrium, NE)状态, 才能使每个机器人利益最大化。纳什平衡状态是指没有任何一个云机器人能够通过改变策略降低自身成本的策略组合, 这就是推算出的最优卸载策略^[22]。

3.1 博弈模型

本文将多个云机器人任务卸载的博弈过程设计成一个博弈模型, 以便找到博弈的NE状态。博弈参与者为云机器人集合 $R = \{1, 2, \dots, i, \dots, n\}$, 云机器人 i 以外的其它云机器人集合为 $R_{-i} = \{1, 2, \dots, i-1, i+1, \dots, n\}$ 。每个云机器人都有策略空间 Q_i , 当前云机器人 i 的任务卸载策略为 $S_i = q_i^k$ ($q_i^k \in Q_i$), 则当前除云机器人 i 以外的其它云机器人的任务卸载策略集合为 $S_{-i} = \{q_1^k, q_2^k, \dots, q_{i-1}^k, q_{i+1}^k, \dots, q_n^k\}$, 全部云机器人的任务卸载策略集合为 $S = \{q_i^k, i \in R\}$ 。当给定其它云机器人的决策 S_{-i} , 云机器人 i 选择合适的卸载策略 S_i 使得自身完成当前计算任务的总成本最小, 即 $\min_{S_i \in Q_i} Cost_i(S_i, S_{-i}), \forall i \in R$ 。由于信道传输速率 r_i 与所有云机器人的卸载策略都有关, 因此云机器人 i 完成当前计算任务的总成本函数可以表示为

$Cost_i(S_i, S_{-i})$, 那么 i 的总成本定义为

$$Cost_i(S_i, S_{-i}) = Cost_i^{\text{loc}} + Cost_i^{\text{mec}} + Cost_i^e \quad (16)$$

根据上述内容, 本文将博弈模型定义为 $G = (R, \{Q_i\}_{i \in R}, \{Cost_i\}_{i \in R})$, 其中 R 是云机器人集合, Q_i 是云机器人 i 的博弈策略集, $Cost_i$ 是评价函数。建模的目标是寻找云机器人最优卸载策略, 据此, 首先给出最佳卸载策略定义。

定义 1: 当给定除云机器人 i 以外其它云机器人策略 S_{-i} 后, 则云机器人 i 的最佳卸载策略为

$$S_i^* = \operatorname{argmin} Cost_i(S_i, S_{-i}) \quad (17)$$

在博弈论中, NE 是一种使得每个参与者的策略都是对于其它参与者策略最优反应的策略组合、是分析多个相互影响的参与者之间决策结果的重要解决方案, 因此, 下边给出 NE 的定义。

定义 2: 纳什平衡策略组合 $S^* = \{S_1^*, S_2^*, \dots, S_i^*, \dots, S_n^*\}$ 是非合作博弈 G 的一个纳什平衡, 则对于 $\forall S_i \in Q_i, i \in R$ 有

$$Cost_i(S_i^*, S_{-i}^*) \leq Cost_i(S_i, S_{-i}^*) \quad (18)$$

纳什平衡状态具有稳定性, 这意味着当所有云机器人的任务卸载策略组合处于 NE 状态时, 云机器人已经不能通过改变自己的卸载策略来达到降低系统代价的目的, 也就说明 NE 状态是一个平衡状态, 没有云机器人有理由背离这个平衡状态, 也就不会再改变各自的卸载策略。

3.2 博弈性质

已知势博弈存在纳什平衡且具有有限改进特性^[23], 下面引入势博弈的概念。

定义 3: 如果博弈模型 $G = (R, \{Q_i\}_{i \in R}, \{Cost_i\}_{i \in R})$ 存在一个势函数 $\psi(i)$ 满足当 $S = \{S_1, S_2, \dots, S_i, \dots, S_n\} \rightarrow S^*$ 时, 对 $\forall i \in R$ 并且 $S_i, S_i' \in Q_i$, 若有 $Cost_i(S_i', S_{-i}) \leq Cost_i(S_i, S_{-i})$, 则有 $\psi(S_i', S_{-i}) \leq \psi(S_i, S_{-i})$, 则称这个博弈为势博弈。

定义 4: 对于有限精确势博弈, 当博弈参与者的策略由 S_i 更新为 S_i' 是代价减小、策略优化的过程, 即 $Cost_i(S_i', S_{-i}) \leq Cost_i(S_i, S_{-i})$, 因此, 将

有限精确势函数描述为

$$\psi(S_i) = \sum_{i=1}^N Cost_i(S_i, S_{-i}) \quad (19)$$

更新策略公式为

$$\Delta\psi(i) = \psi(S_i) - \psi(S_i') \quad (20)$$

$\Delta\psi(i)$ 的值越大则说明云机器人更新策略的机会越大, 因此需要找出 $\text{argmax}\Delta\psi(i)$, $i \in R$, 此时云机器人 i 的最佳卸载策略更新为对应的策略 S_i^* , 在经过有限的更新迭代后博弈达到 NE 状态, 进而找到纳什平衡策略组合 S^* 。

3.3 任务卸载策略算法设计

本文设计云边协同下基于博弈论的部分任务卸载策略算法(game theory-partial task offloading, GT-PTO)。为充分利用云服务器与边缘云服务器的计算资源, 将云机器人当前任务进行部分分布式卸载。借助于把多个云机器人任务通过同一信道卸载的博弈过程, 进而找到所有云机器人卸载任务量的纳什平衡状态, 从而优化云机器人完成计算任务的代价。

GT-PTO 算法设计: 首先, 将所有云机器人的任务卸载策略初始化为当前计算任务本地计算, 即将策略组合初始化为 $S = \{S_i = (1, 1), i \in R\}$, 其次, 在 S_{-i} 不变的情况下依次更新云机器人 i 的卸载策略 S_i' , 对比找到差值最大的 $\text{argmax}\Delta\psi(i)$, $i \in R$ 的云机器人 i 以及其相对应的最佳卸载策略 S_i^* , 并更新策略组合 S 。再次, 根据更新后的策略组合更新任务卸载上行传输速率 r_i 。重复上述策略更新过程, 依次找到每个云机器人的最佳卸载策略, 最终更新的策略组合 S 就是纳什平衡策略组合 S^* 。具体 GT-PTO 算法设计如下:

输入: 云机器人 R , 计算任务 τ_i , 边缘云及云服务器计算资源 F_{mec} 和 F_c , 信道带宽 B , 传输功率 p_i 以及阈值划分精度 ζ 等初始数据。

输出: 云机器人最优策略组合 S^* , 系统最小总成本开销 $\sum_{i=1}^N Cost_i(S^*)$ 。

初始化: 每个云机器人本地执行, 策略组合 $S = \{S_i = (1, 1), i \in R\}$ 。

while $S = S^*$:

for $i = 1$ to N do:

for $k = 2$ to K do:

根据当前策略组合 S 更新信道传输速率 r_i ;

根据当前策略 q_i^k 及式(15)计算云机器人的任务成本 $Cost_i(q_i^k)$;

if $Cost_i(q_i^k) < Cost_i(q_i^{k-1})$:

$S_i' = q_i^k$

else:

$S_i' = q_i^{k-1}$

end for

选择出最佳卸载策略 $S_i^* = \text{argmin} Cost_i(S_i, S_{-i})$ 但不更新策略组合 S ;

根据式(19)和(20)算出 i 的 $\text{argmax}\Delta\psi(i)$;

end for

所有云机器人争夺决策更新机会;

比较所有 $\text{argmax}\Delta\psi(i)$, 找到其中差值最大的对应的云机器人 i , 更新其卸载策略为 $S_i^* = \text{argmin} Cost_i(S_i, S_{-i})$, 并更新策略组合 S ;

continue

if $S = S^*$:

计算当前策略组合下的系统总成本开销

$$\sum_{i=1}^N Cost_i(S^*);$$

end if

其中 $S^* = \{S_1^*, S_2^*, \dots, S_i^*, \dots, S_N^*\}$, 每一次循环可以更新出一个云机器人 i 对应的最佳卸载策略 S_i^* , 经过有限次循环后找到纳什平衡策略组合 S^* 。

4 实验仿真

4.1 实验设置

仿真实验在 Pycharm 平台利用 Python 语言编程实现。采用光纤无线混合网络下集中式云服务和 MEC 协同系统, 实验主要参考文献[15, 20]设置

无线通信和重要计算参数。如表1所示。

表1 实验参数
Table 1 Experimental parameters

参数	数值
系统带宽 B/MHz	1
上行传输功率 p_i/mW	100
信道噪声功率 ω_0/dBm	-100
能耗系数 η	0.5×10^{-25}
BS到云的单位数据传输时延 $v/(\mu\text{s/bit})$	1
MEC服务器计算能力 $F_{\text{mec}}/\text{GHz}$	5
CCC服务器计算能力 F_c/GHz	20

云边系统中云机器人数量 N 从 $[5, 30]$ 中进行选取，云机器人的计算能力从 $\{0.5, 0.7, 1.0\}$ GHz (CPU运行的时钟频率) 集合中随机选择。信道增益 $h_i = d_i^{-\sigma}$ ， d_i 为云机器人 i 到基站之间的距离，设置为 100 M； σ 是衰减系数，设置 $\sigma = 4$ 。考虑提出的 GT-PTO 是基于博弈论的云边端协同卸载策略，因此，部分实验与传统博弈论策略进行了比较。

本文考虑多个异构云机器人计算任务不尽相同，参考文献[24]，输入计算任务 τ_i 的数据量 D_i 从 500~1 000 kB 中随机分配，每个任务对应的计算量 C_i 从 200~500 MC(完成计算任务需要的CPU周期数量)中随机选取。返回计算结果的数据量 D_i' 设置为计算数据量的 5%。

4.2 实验分析

图2是多个云机器人在云边协同环境下不同卸载策略的系统总代价对比。由图2可知，随着云机器人数量的不断增加，各种卸载策略下的系统总代价都在逐渐增加。相比于将计算任务全部本地执行、全部卸载到边缘服务器(MEC)端或全部卸载到云中心端(CCC)执行，云边环境下传统的 Game Theory 算法能更有效的降低系统代价。但此时算法仅仅考虑将每个云机器人的单个任务找到其最佳执行位置，并求出系统最优总代价，并未考虑任务的可切割性，而本文提出的 GT-PTO 算法，通过将每个云机器人的单个任务分割成三部分，在本地、MEC 服务器、CCC 服务器分别执行

单个任务的部分任务，通过博弈论的方法找到任务分割的最佳卸载策略阈值，从而达到系统总代价的进一步优化。由图2可以发现：当云机器人数量不断增加，GT-PTO 算法对于系统总代价的优化效果越来越好。这也进一步验证了本文卸载策略在云边协同环境下有更好的性能。

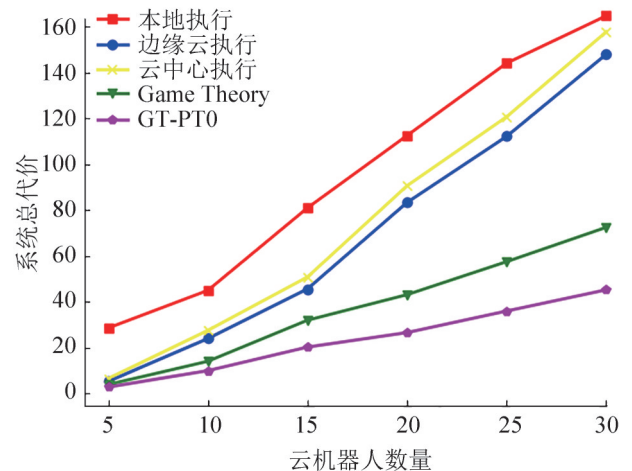


图2 不同卸载策略下的系统总代价比较

Fig. 2 Comparison of total system cost under different offloading strategies

图3是 GT-PTO 算法中不同阈值划分精度下产生的系统总代价对比图。根据3图中数据分析可知，当阈值划分步长 ζ 给定后，可以计算出其对应的策略空间 Q_i 的大小。在实验中给定的 ζ 分别为 0.1、0.05 和 0.02，根据 2.1 节公式 $Q_i = \{(\alpha_i^k, \beta_i^k)\}$ 其对应的策略空间大小分别为 66、231 和 1 326。当 ζ 设置为 0.01 时云机器人任务卸载的策略空间已经变得非常大，但是其能达到的优化效果并不会太大的提升。因此，本文将阈值精度设置 2%、5% 和 10%，观察在这个三种不同的阈值精度下的系统总代价对比。由图3可知， ζ 越小，阈值精度越高，其对应的策略空间也越大，在同一组云机器人的随机数据下完成计算任务产生的系统总代价也会越小，但其并不是决定性因素，只需设置合理的阈值精度即可。因此，在后续实验中，本文统一使用阈值精度 5% 作为实验常量。

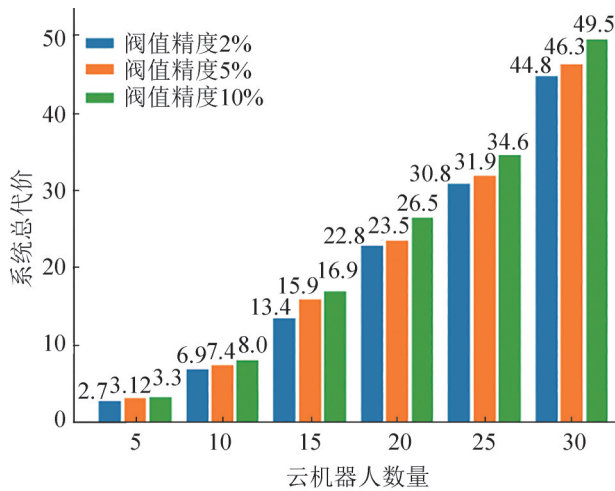
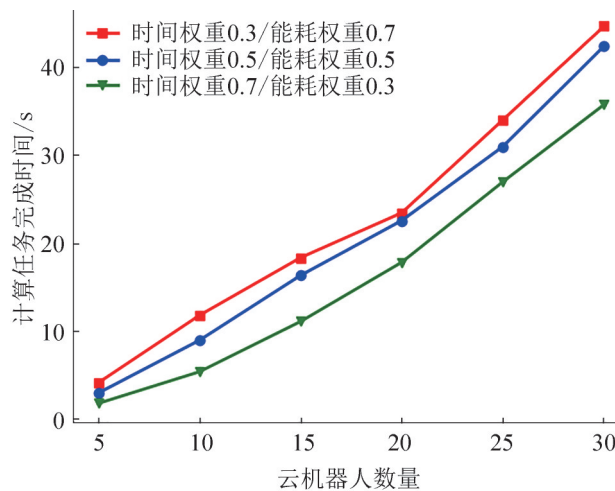
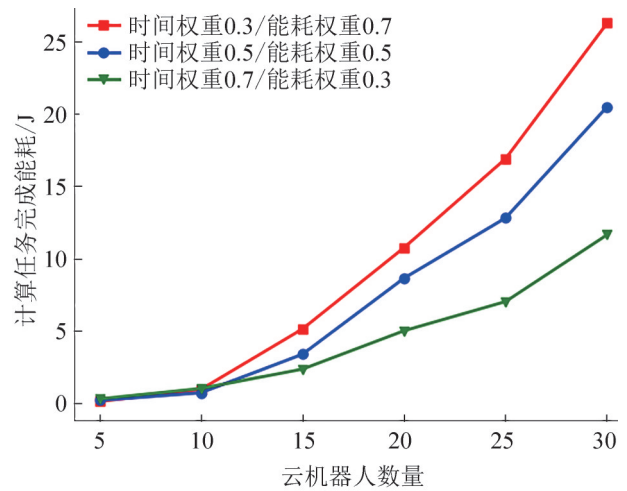


图3 GT-PTO算法下采用不同阈值精度系统总代价的比较
Fig. 3 Comparison of total system cost under different threshold precision with GT-PTO algorithm

图4是GT-PTO算法中时间与能耗权重分别对计算任务完成时间与能耗的影响对比。由图4(a)可以分析出: 无论时间权重占比多大, 随着云机器人数量不断增加, 系统完成任务的时间呈线性增加的趋势。但在同一云机器人数量状态下时间权重越大, 所需任务完成时间越少。图4(b)中, 系统完成计算任务总能耗随云机器人数量增加而增加, 但在同一云机器人数量系统中能耗权重越大, 计算任务完成的系统总能耗越低。两个实验对比图符合不同权重下系统完成任务产生的总能耗和时间占比不同的理论数据。



(a) 时间权重对任务完成时间的影响



(b) GT-PTO算法下能耗权重对任务计算能耗的影响

图4 权重对任务的影响
Fig. 4 Influence of weights on tasks

图5是云机器人本地计算能力对系统总代价的影响实验。在准静态场景且同一组实验参数设置下, 所有云机器人本地计算能力分别设置为0.5, 0.7, 1.0 GHz。在三种不同状态下分析实验数据可知, 在本文参数设置下本地计算能力越强最终系统完成所有计算任务总代价反而越大, 这是因为本地计算能力会影响NE状态下云机器人计算任务的卸载量, 从而影响整体完成任务的总代价。

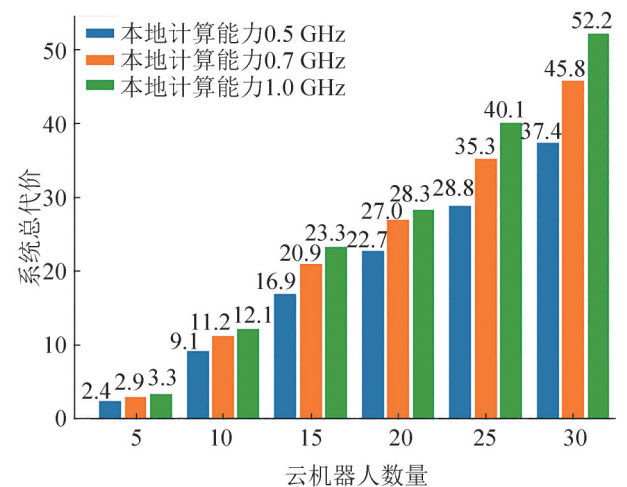


图5 GT-PTO算法下本地计算能力对系统总代价的影响
Fig. 5 Influence of local computing power on total system cost

为测试本地计算能力对计算任务卸载量的影响,本文设计了相关测试实验,如图6所示,并增加了0.8, 0.9 GHz的机器人本地计算能力的表示,探知在本文算法下达到NE状态时各个云机器人任务卸载量的大小。分析图6可知,大体上云机器人计算能力越强,计算任务卸载量会越多,这是因为本地完成计算任务时计算能力越强产生的时延虽然变小,但本地能耗会产生更多,因此需要将更多的计算任务卸载到MEC或云端进行完成。影响云机器人任务卸载量的因素很多,例如代价权重,通信带宽等,因此在现实场景中可以通过改变这些变量,以达到不同需求的目的。

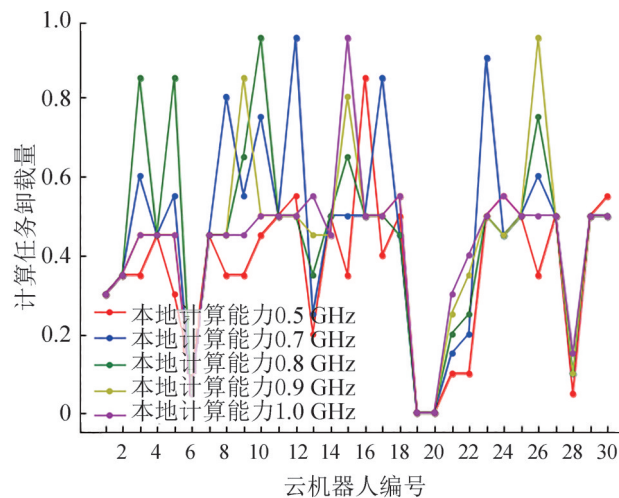


图6 纳什平衡状态下本地计算能力对云机器人任务卸载量的影响

Fig. 6 Influence of local computing power on task offloading volume of cloud robots in Nash equilibrium state

5 结论

本文研究了多云机器人在云边环境下进行计算任务协同卸载的问题,以最大程度的降低系统完成任务代价为目标,提出了云边环境下基于博弈论的部分任务卸载策略算法(GT-PTO)。仿真实验表明,在当前系统下本算法拥有良好的性能,可以有效降低系统完成任务的总代价。当然,本研究仍存在一定局限性。如,目前系统中仅考虑了单信道,单MEC服务器与云服务器协同系统,

而未考虑在更复杂的场景中进行任务卸载的情况,如多信道,多MEC服务器与云服务器协同系统。今后的研究方向会考虑将GT-PTO算法带入到更复杂的云边环境下,进一步优化云机器人卸载的过程,减少云机器人完成计算任务的系统总代价。

参考文献:

- [1] Miratabzadeh S A, Gallardo N, Gamez N, et al. Cloud Robotics: Software Architecture for Heterogeneous Large-Scale Autonomous Robots[C]// 2016 World Automation Congress (WAC). Rio Grande, PR, USA: IEEE, 2016.
- [2] Saha O, Dasgupta P. A Comprehensive Survey of Recent Trends in Cloud Robotics Architectures and Applications[J]. Robotics (S2218-6581), 2018, 7(3): 47.
- [3] Wang Lujia, Ming Liu, Meng M Q. A Hierarchical Auction-Based Mechanism for Real-Time Resource Allocation in Cloud Robotic Systems[J]. IEEE Transactions on Cybernetics (S2168-2267), 2016, 47(2): 473-484.
- [4] Wang L, Liu M, Meng M Q. A Pricing Mechanism for Task Oriented Resource Allocation in Cloud Robotics[C]// Robots and Sensor Clouds. Switzerland: Springer Cham, 2016: 3-31.
- [5] Kong Yan, Zhang Minjie, Ye Dayong. A Belief Propagation-Based Method for Task Allocation in Open and Dynamic Cloud Environments[J]. Knowledge-Based Systems (S0950-7051), 2017 (115): 123-132.
- [6] Abbas N, Zhang Y, Taherkordi A, et al. Mobile Edge Computing: A Survey[J]. IEEE Internet of Things Journal (S2327-4662), 2017, 5(1): 450-465.
- [7] Ahmed E, Rehmani M H. Mobile Edge Computing: Opportunities, Solutions, and Challenges[J]. Future Generation Computer Systems (S0167-739X), 2016, 70 (5): 59-63.
- [8] Yang L, Cao J, Cheng H, et al. Multi-user Computation Partitioning for Latency Sensitive Mobile Cloud Applications[J]. IEEE Transactions on Computers (S0018-9340), 2015, 64(8): 2253-2266.
- [9] C You, K Huang, H. Chae, et al. Energy-Efficient Resource Allocation for Mobile-edge Computation Offloading[J]. IEEE Transactions on Wireless Communications (S1536-1276), 2016, 16(3): 1397-1411.
- [10] Dai Y, Xu D, Maharjan S, et al. Joint Computation Offloading and User Association in Multi-task Mobile Edge Computing[J]. IEEE Transactions on Vehicular Technology (S0018-9545), 2018(67): 12313-12325.

- [11] Guo J L, Wu J G, Chen L, et al. A Collaborative Computation Offloading Algorithm Based on Fiber-Wireless Networks[J]. *Computer Engineering and Science* (S2231-6590), 2019, 41(1): 31-40.
- [12] Guo H, Liu J. Collaborative Computation Offloading for Multiaccess Edge Computing Over Fiber-Wireless Networks[J]. *IEEE Transactions on Vehicular Technology* (S0018-9545), 2018, 67(5): 4514-4526.
- [13] Xu C, Lei J, Li W, et al. Efficient Multi-user Computation Offloading for Mobile-Edge Cloud Computing[J]. *IEEE/ACM Transactions on Networking* (S1063-6692), 2016, 24(5): 2795-2808.
- [14] Jie Y, Tang X, Raymond K K, et al. Online Task Scheduling for Edge Computing Based on Repeated Stackelberg Game[J]. *Journal of Parallel and Distributed Computing* (S0743-7315), 2018(122): 159-172.
- [15] Zhou S, Waqas J. The Partial Computation Offloading Strategy Based on Game Theory for Multi-user in Mobile Edge Computing Environment[J]. *Computer Networks* (S1389-1286), 2020(178): 107334.
- [16] 吴学文, 廖婧贤. 云边协同系统中基于博弈论的资源分配与任务卸载方案[J]. *系统仿真学报*: 2022, 34(1): 1-13.
Wu Xuwen, Liao Jingxian. Game-Based Resource Allocation and Task Offloading Scheme in Collaborative Cloud-edge Computing System[J]. *Journal of System Simulation*, 2022, 34(1): 1-13.
- [17] Yang L, Cao J, Yuan Y, et al. A Framework for Partitioning and Execution of Data Stream Applications in Mobile Cloud Computing[J]. *ACM SIGMETRICS Performance Evaluation Review* (S0163-5999), 2013, 40(4): 23-32.
- [18] Chen Xu. Decentralized Computation Offloading Game for Mobile Cloud Computing[J]. *IEEE Transactions on Parallel and Distributed Systems* (S1045-9219), 2014, 26(4): 974-83.
- [19] Hu J, Li K, Liu C, et al. Game-Based Task Offloading of Multiple Mobile Devices with QoS in Mobile Edge Computing Systems of Limited Computation Capacity[J]. *ACM Transactions on Embedded Computing Systems* (S1558-3465), 2020, 19(4): 1-21.
- [20] Xu F, Yang W X, Li H. Computation Offloading Algorithm for Cloud Robot Based on Improved Game Theory[J]. *Computers & Electrical Engineering* (S0049-7506), 2020, 87: 106764.
- [21] Yang L, Cao J, Cheng H, et al. Multi-user Computation Partitioning for Latency Sensitive Mobile Cloud Applications[J]. *IEEE Transactions on Computers* (S0018-9340), 2015, 64(8): 2253-2266.
- [22] 杨卫霞. 基于博弈论的云机器人计算卸载策略研究[D]. 西安: 西安工业大学, 2020.
Yang Weixia. Research on Cloud Robot Computing Offloading Strategy Based on Game Theory[D]. Xi'an: Xi'an Technological University, 2020.
- [23] 王艺洁, 凡佳飞, 王陈宇. 云边环境下基于博弈论的两阶段任务迁移策略[J]. *计算机应用*, 2021, 41(5): 1392-1398.
Wang Yijie, Fan Jiafei, Wang Chenyu. Two-stage Task Offloading Strategy Based on Game Theory in Cloud-Edge Environment[J]. *Journal of Computer Applications*, 2021, 41(5): 1392-1398.
- [24] Ren J, Yu G, He Y, et al. Collaborative Cloud and Edge Computing for Latency Minimization[J]. *IEEE Transactions on Vehicular Technology* (S0018-9545), 2019, 68(5): 5031-5044.