

2-16-2023

Computation Offloading Strategy Based on Stackelberg Game and DRL

Xianwei Zhou

School of Software, South China Normal University, Foshan 528225, China,; 20871147@qq.com

Qixu Gong

School of Software, South China Normal University, Foshan 528225, China;

Songsen Yu

School of Software, South China Normal University, Foshan 528225, China,; yss8109@163.com

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.

Computation Offloading Strategy Based on Stackelberg Game and DRL

Abstract

Abstract: To achieve the optimal computation offloading strategy for two kinds of MEC users in 5G hybrid private network, Stackelberg game is used to build the model of the competition for MEC server resources of two kinds of users, and the strategies of complete information game and partially incomplete information game are researched respectively. *It is proved that there is only one Nash equilibrium solution in the complete information scenario. In the incomplete information scenario, the environment is modeled as POMDP, and a two-stage deep reinforcement learning (TSDRL) is proposed to obtain the optimal computation offloading strategy.* Simulation results show the proposed algorithm having a total reduction of 20.81% time delay and 3.38 % energy consumption compared with the D-DRL algorithm and can effectively improve the user QoE (quality of experience).

Keywords

5G hybrid private network, computation offloading, Stackelberg game theory, Nash equilibrium, partially observable Markov decision process (POMDP)

Recommended Citation

Xianwei Zhou, Qixu Gong, Songsen Yu. Computation Offloading Strategy Based on Stackelberg Game and DRL[J]. Journal of System Simulation, 2023, 35(2): 372-385.

基于 Stackelberg 博弈与深度强化学习的计算卸载策略

周娴玮, 龚启旭, 余松森*

(华南师范大学 软件学院, 广东 佛山 528225)

摘要: 为使 5G 混合专网结构的 2 种用户能获得最优计算卸载策略, 将 2 种用户竞争移动边缘计算 (mobile edge computing, MEC) 服务器资源的问题建模成 Stackelberg 博弈, 并分别讨论了完全信息博弈和不完全信息博弈下的策略。完全信息博弈下, 存在唯一纳什均衡解; 不完全信息博弈下, 将环境建模为部分可观测的马尔可夫决策过程 (partially observable Markov decision process, POMDP), 并提出一种基于二阶段深度强化学习 (two-stage deep reinforcement learning, TSDRL) 的最优卸载策略。仿真实验表明: 该算法相较于 D-DRL 算法能减少 20.81% 的时延及 3.38% 的能耗, 有效提高用户 QoE (quality of experience)。

关键词: 5G 混合专网; 计算卸载; Stackelberg 博弈; Nash 均衡; 马尔可夫决策过程

中图分类号: TP393.01

文献标志码: A

文章编号: 1004-731X(2023)02-0372-14

DOI: 10.16182/j.issn1004731x.joss.21-1118

引用格式: 周娴玮, 龚启旭, 余松森. 基于 Stackelberg 博弈与深度强化学习的计算卸载策略[J]. 系统仿真学报, 2023, 35(2): 372-385.

Reference format: Zhou Xianwei, Gong Qixu, Yu Songsen. Computation Offloading Strategy Based on Stackelberg Game and DRL[J]. Journal of System Simulation, 2023, 35(2): 372-385.

Computation Offloading Strategy Based on Stackelberg Game and DRL

Zhou Xianwei, Gong Qixu, Yu Songsen*

(School of Software, South China Normal University, Foshan 528225, China)

Abstract: To achieve the optimal computation offloading strategy for two kinds of MEC users in 5G hybrid private network, Stackelberg game is used to build the model of the competition for MEC server resources of two kinds of users, and the strategies of complete information game and partially incomplete information game are researched respectively. It is proved that there is only one Nash equilibrium solution in the complete information scenario. In the incomplete information scenario, the environment is modeled as POMDP, and a two-stage deep reinforcement learning (TSDRL) is proposed to obtain the optimal computation offloading strategy. Simulation results show the proposed algorithm having a total reduction of 20.81% time delay and 3.38% energy consumption compared with the D-DRL algorithm and can effectively improve the user QoE (quality of experience).

Keywords: 5G hybrid private network; computation offloading; Stackelberg game theory; Nash equilibrium; partially observable Markov decision process (POMDP)

收稿日期: 2021-11-02 修回日期: 2022-01-07

基金项目: 广东省应用型科技研发重大专项(2016B020244003); 广东省基础与应用基础研究基金(2020B1515120089, 2020A1515110783); 广东省企业科技特派员项目(GDKTP2020014000)

第一作者: 周娴玮(1982-), 男, 讲师, 博士, 研究方向为机器人技术、多传感信息融合。E-mail: 20871147@qq.com

通讯作者: 余松森(1972-), 男, 教授, 博士, 研究方向为计算机视觉、物联网技术。E-mail: yss8109@163.com

0 引言

5G网络逐渐遍布人们的生活, 基站建设成本高、功耗大的问题也日益凸显。为了降成本, 同时能保持工业用户高效工作, 行业提出了5G混合专网结构^[1]。该结构以5G数据分流技术为基础, 通过无线和控制网元的灵活定制, 以网络切片的形式将用户分为专网用户和公网用户^[2]。5G混合专网中, 边缘计算作为一门新范式被引入, 其将计算能力放置在边缘服务器上, 从而降低服务时延^[3-4]。

目前, 针对边缘计算卸载主要集中在如何通过任务卸载^[5]、资源分配^[6]来减少传输时延^[7]或节约终端能量^[8]来提高用户的QoE(quality of experience)。You等^[9]将计算卸载问题描述为在计算延迟的约束下最小化各移动终端用户能量消耗的加权总和凸优化问题。Chen等^[10]在移动边缘计算(mobile edge computing, MEC)多用户系统中, 考虑全局的能量消耗与执行时延最小化的联合目标。文献[11]考虑用户具有能量采集的特性, 提出了李雅普诺夫贪婪优化的算法来降低全局的时延与能耗的联合成本, 从而提高用户的Qos(quality of service)。

上述工作都需要集中控制, 但普通用户传输速率、能耗等信息很难被全局掌控。最近的一些研究使用博弈论的分布式方案来解决多用户计算卸载问题。在分布式环境下, Tran等^[12]提出了离散随机策略的多智能体纳什Q学习。Chao等^[13]提出了多智能体马尔可夫决策过程的DDPG方法, 其中环境是完全可观测的。Zhan等^[14]将博弈建模为部分可观测的马尔可夫决策过程(partially observable Markov decision process, POMDP), 并检查了当前策略梯度和Actor-Critic算法的作用, 结合DNC网络和强化学习算法提出了面向多用户博弈的分布式强化学习(decentralized approach based on deep reinforcement learning, D-DRL)算法。文献[15]提出了一种采用贝叶斯博弈和深度强化学

习结合的方法, 使矿工在2种区块链下的边缘计算得到最优卸载, 最终证明了贝叶斯和深度强化学习算法能达到最优贝叶斯均衡。Li等^[16]提出了一种基于Stackelberg博弈的双稳态计算卸载机制, 用于分析多个边缘云与多个终端设备之间的定价博弈, 提出了基于社会互动信息制定的效用函数, 实现了Stackelberg博弈下多用户的最优卸载。

然而, 以上文献并未针对混合专网的网络架构进行探讨, 只针对同类型的卸载用户进行研究, 在面对不同网络切片环境下的用户以及对性能需求不同的用户时, 以上文献的算法很难做到合理的计算卸载。文献[17]对公私2种用户分别设立了独立的MEC服务器, 此种方案虽然具有更高的稳定性, 但是对资源利用不充分。文献[18]提出了云边协同系统下的博弈论方案, 该方案能有效降低时延和能耗, 还可以感知不同用户的优先级, 优先处理紧急用户任务, 然而该方案优先级是通过设定优先级系数来实现, 人为系数设定困难且对时变网络作用有限。

针对这一情况, 本文研究了专用网络用户和公用网络用户共用一个MEC服务器时的计算卸载与资源分配问题。并将该主从结构的博弈问题建模成Stackelberg博弈模型, 考虑博弈中不完全信息博弈的情况, 提出了基于二阶段深度强化学习(two-stage deep reinforcement learning, TSDRL)算法。在优先考虑数据量大的专网用户情况下相较于以往的博弈方法, 能有效的降低用户的时延和能耗。

1 系统模型和问题描述

计算卸载考虑专网用户 i 和公网用户 j , 2种用户共同竞争MEC服务器资源, 如图1所示。

1.1 5G混合双网通信模型

在博弈系统中, 所有用户都可以自己独立做出决策, 将自己的任务进行卸载。对于专网用户, 其计算卸载量表示为 $x_{1,i}^k$, 其中 k 表示时隙, 每个用户最大数据量为 $C_{1,i}$, 因此有 $x_{1,i}^k \in [0, C_{1,i}]$ 。同理

对于公网用户，其计算卸载量表示为 $x_{f,j}^k$ 。专网用户和公网用户的决策空间分别为 $\mathbf{X}_1 = \prod_{i=1}^N X_{1,i}$ 和 $\mathbf{X}_f = \prod_{j=1}^M X_{f,j}$ ，数据卸载向量分别为 $\mathbf{x}_1 = \{x_{1,1}, x_{1,2}, \dots, x_{1,N}\} \in \mathbf{X}_1$ 和 $\mathbf{x}_f = \{x_{f,1}, x_{f,2}, \dots, x_{f,M}\} \in \mathbf{X}_f$ ，其中， N 为专网用户数量， M 为公网用户数量。

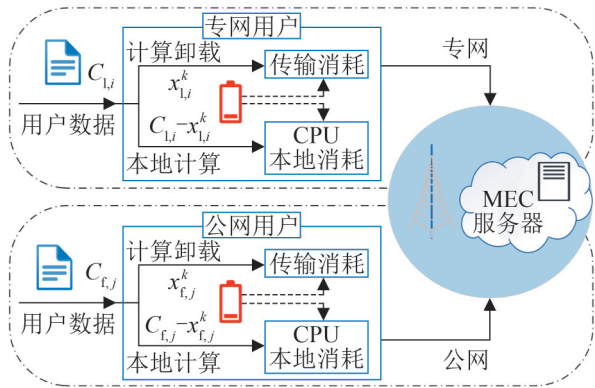


图1 5G混合专网下不同用户的卸载
Fig. 1 Offloading of different users in 5G hybrid private network

在混合专网的拓扑网络中，专网用户和公网用户看成一个个终端设备，根据香农第二定理，以专网用户为例，可以得到信道传输速率为

$$r_{1,i} = w_{1,i} \text{lb} \left(1 + \frac{\rho_{1,i} |h_{1,i}|^2}{\sigma^2} \right) \quad (1)$$

式中： $\rho_{1,i}$ 为传输功率； $w_{1,i}$ 为信道宽度； $h_{1,i}$ 为信道增益； σ 为噪声。

1.2 效用函数

MEC服务器的计算资源根据用户上传的数据量大小进行分配，假设服务器的总计算资源为 R ，那么当一个用户卸载数据量 x 所占用的计算资源为 $\frac{xR}{\sum_{i=1}^N x_{1,i}^k + \sum_{j=1}^M x_{f,j}^k}$ ^[16,19]。以专网用户为例，

用户完成任务的时延为计算卸载时延与本地计算时延的最大取值，即 $t_{1,i}^k = \max \left\{ \frac{x_{1,i}^k}{r_{1,i}^k} + \right.$

$$\left. \frac{\sum_{i=1}^N x_{1,i}^k + \sum_{j=1}^M x_{f,j}^k}{R}, \frac{C_{1,i} - x_{1,i}^k}{F_{1,i}} \right\}$$

其中 $F_{1,i}$ 为本地计算资源。令 $p_{1,i}$ 表示单位时间传输的成本，那么对于卸载数据 $x_{1,i}^k$ ，传输成本为 $p_{1,i} \frac{x_{1,i}^k}{r_{1,i}^k}$ 。考虑到终端设备

在本地运行时消耗设备的电源能量，假定单位数据能耗成本为 $v_{1,i}$ ，那么本地计算所消耗成本可表示为 $v_{1,i}(C_{1,i} - x_{1,i}^k)$ 。由于专网用户的数据量极大，而服务器的总资源有限，为了防止普通用户完全竞争不到计算资源，导致其宕机，因此，专网用户需要额外考虑公网用户的卸载量。据此，可以得到专网用户 i 基于服务器资源成本、传输成本、本地能耗成本在时隙 k 的效用函数为

$$u_{1,i}(x_{1,i}^k, \mathbf{x}_{1,-i}^k, \mathbf{x}_f^k) = \frac{(\alpha_{1,i} x_{1,i}^k + \beta_{1,i} \sum_{j=1}^M x_{f,j}^k) R}{\sum_{i=1}^N x_{1,i}^k + \sum_{j=1}^M x_{f,j}^k} - \left(p_{1,i} \frac{x_{1,i}^k}{r_{1,i}^k} + v_{1,i} (C_{1,i} - x_{1,i}^k) \right) \quad (2)$$

其中， $\mathbf{x}_{1,-i}^k = \{x_{1,1}^k, x_{1,2}^k, \dots, x_{1,i-1}^k, x_{1,i+1}^k, \dots, x_{1,N}^k\}$ ， $\alpha_{1,i}$ 和 $\beta_{1,i}$ 分别表示该用户的卸载倾向以及对其他公网用户的态度权重，由于专网用户对自身优先级高于公网用户，那么有 $\alpha_{1,i} > \beta_{1,i}$ 。效用函数用来衡量用户在资源博弈过程中的满足程度，其中，资源成本通过系数 $\alpha_{1,i}$ 和 $\beta_{1,i}$ 、传输成本通过 $p_{1,i}$ 、能耗成本通过 $v_{1,i}$ 进行归一化。该效用函数也可以通过增加项来衡量现实中用户复杂场景，如抵御网络风险^[20]、用户互动信息^[16]等。

对于公网用户，主要竞争专网用户竞争完之后的剩余资源，并且在专网用户每次竞争完之后，服务器会向公网用户广播专网用户使用后的服务器资源剩余情况。此时该博弈可以看成是公网用户之间博弈，即为古诺模型。因此，对于任一公网用户，当卸载数据量为 $x_{f,j}^k$ 时，所占用MEC服务器计算资源为

$$R_f = \frac{\beta_{f,j} x_{f,j}^k}{\sum_{j=1}^M x_{f,j}^k} \left(R - \frac{\sum_{i=1}^N x_{l,i}^k}{\sum_{i=1}^N x_{l,i}^k + \sum_{j=1}^M x_{f,j}^k} R \right) = \frac{\beta_{f,j} x_{f,j}^k}{\sum_{i=1}^N x_{l,i}^k + \sum_{j=1}^M x_{f,j}^k} R \quad (3)$$

在时隙 k 时, 公网用户 j 效用函数为

$$u_{f,j}(x_{f,j}^k, \mathbf{x}_{f,-j}^k, \mathbf{x}_1^k) = \frac{\beta_{f,j} x_{f,j}^k R}{\sum_{j=1}^M x_{f,j}^k + \sum_{i=1}^N x_{l,i}^k} - \left(p_{f,j} \frac{x_{f,j}^k}{r_{f,j}^k} + v_{f,j} (C_{f,j} - x_{f,j}^k) \right) \quad (4)$$

式中: $\mathbf{x}_{f,-j}^k = \{x_{f,1}^k, x_{f,2}^k, \dots, x_{f,j-1}^k, x_{f,j+1}^k, \dots, x_{f,M}^k\}$ 。

1.3 问题描述

已知专网用户和公网用户的效用函数, 对于所有的专网用户, 其目标为使其获得最大满意程度, 可描述为一个效用函数最大化的问题:

$$\begin{aligned} \max u_{l,i}(x_{l,i}^k, \mathbf{x}_{l,-i}^{k*}, \mathbf{x}_f^{k*})_{i \in N} \quad (5) \\ \text{s.t. } C_1: \mathbf{x}_f^{k*} = \arg \max \{u_f(\mathbf{x}_f^{k*})\}, \\ C_2: 0 < x_{l,i}^{k*} < C_{l,i}, i \in \{1, 2, \dots, N\} \\ C_3: 0 < x_{f,j}^{k*} < C_{f,j}, j \in \{1, 2, \dots, M\} \end{aligned}$$

其中, $\mathbf{x}_{l,-i}^{k*} = \{x_{l,1}^{k*}, x_{l,2}^{k*}, \dots, x_{l,i-1}^{k*}, x_{l,i+1}^{k*}, \dots, x_{l,N}^{k*}\}$, $\mathbf{x}_f^{k*} = \{x_{f,1}^{k*}, x_{f,2}^{k*}, \dots, x_{f,M}^{k*}\}$, 同理, 对于公网用户的优化问题也可以类似描述。所以无论是专网用户还是公网用户, 都是要实现效用最大化。据此, 定义以下为 Stackelberg 博弈的纳什均衡点。

定义 1: 假定 \mathbf{x}_1^{k*} 为专网用户最优卸载策略, \mathbf{x}_f^{k*} 为公网用户最优卸载策略, 那么满足式(6)的点 $(\mathbf{x}_1^{k*}, \mathbf{x}_f^{k*})$ 称为 Stackelberg 博弈纳什均衡点:

$$\begin{aligned} u_{l,i}(x_{l,i}^{k*}, \mathbf{x}_{l,-i}^{k*}, \mathbf{x}_f^{k*}) \geq u_{l,i}(x_{l,i}^k, \mathbf{x}_{l,-i}^{k*}, \mathbf{x}_f^{k*})_{i \in N} \quad \&\& \\ u_{f,j}(x_1^{k*}, x_{f,j}^{k*}, \mathbf{x}_{f,-j}^{k*}) \geq u_{f,j}(x_1^{k*}, x_{f,j}^k, \mathbf{x}_{f,-j}^{k*})_{j \in M} \quad (6) \end{aligned}$$

定义 1 的博弈称为多主多从的 Stackelberg 博弈模型^[21], 当所有用户都选择最优卸载策略时, 该博弈达到纳什均衡。纳什均衡具有良好的自稳定特性, 使得处于均衡状态的用户可以得到一个

双方都满意的解, 而没有用户有偏离的动机。因为用户可以按照自己的收益行动^[14], 所以这个属性对于分布式计算卸载问题非常重要。本文的问题为求解该 Stackelberg 模型下的纳什均衡, 即所有用户的最优卸载策略。

2 完全信息博弈下卸载策略求解

假定所有用户的信息是共享的, 所谓的信息共享指的是所有的用户知道彼此的信道传输速率、能耗系数等信息。

2.1 纳什均衡唯一性分析

对于专网用户, 由式(2)对专网用户效用函数求导:

$$\frac{\partial u_{l,i}}{\partial x_{l,i}^k} = \frac{\alpha_{l,i} R}{\sum_{i=1}^N x_{l,i}^k + \sum_{j=1}^M x_{f,j}^k} - \frac{(\alpha_{l,i} x_{l,i}^k + \beta_{l,i} \sum_{j=1}^M x_{f,j}^k) R}{\left(\sum_{i=1}^N x_{l,i}^k + \sum_{j=1}^M x_{f,j}^k \right)^2} - \frac{p_{l,i}/r_{l,i}^k + v_{l,i}}{\quad} \quad (7)$$

再进行求二阶导:

$$\frac{\partial^2 u_{l,i}}{\partial (x_{l,i}^k)^2} = -2R \frac{\alpha_{l,i} \sum_{n \neq i}^N x_{l,n}^k + (\alpha_{l,i} - \beta_{l,i}) \sum_{j=1}^M x_{f,j}^k}{\left(\sum_{i=1}^N x_{l,i}^k + \sum_{j=1}^M x_{f,j}^k \right)^3} < 0 \quad (8)$$

由于 MEC 服务器计算资源 $R > 0$, 卸载策略 $x_{l,i}^k > 0$, 2 种用户权重系数 $\alpha_{l,i} > \beta_{l,i}$, 得专网用户的二阶导数 < 0 , 因此, 该函数是严格的凸函数, 即存在唯一纳什均衡点。那么, 如果最优卸载策略 $x_{l,i}^{k*}$ 存在, 那么它必定是唯一的。令一阶导数 $\frac{\partial u_{l,i}}{\partial x_{l,i}^k} = 0$, 得

$$\frac{\alpha_{l,i} R}{\sum_{i=1}^N x_{l,i}^k + \sum_{j=1}^M x_{f,j}^k} - \frac{(\alpha_{l,i} x_{l,i}^k + \beta_{l,i} \sum_{j=1}^M x_{f,j}^k) R}{\left(\sum_{i=1}^N x_{l,i}^k + \sum_{j=1}^M x_{f,j}^k \right)^2} - \frac{p_{l,i}}{r_{l,i}^k} + v_{l,i} = 0 \quad (9)$$

解方程(9)，可得

$$\tilde{x}_{l,i}^k = \sqrt{\frac{\alpha_{l,i} \sum_{n \neq i}^N x_{l,n}^k + (\alpha_{l,i} - \beta_{l,i}) \sum_{j=1}^M x_{f,j}^k}{\frac{p_{l,i}}{r_{l,i}^k} - v_{l,i}} R - \left(\sum_{n \neq i}^N x_{l,n}^k + \sum_{j=1}^M x_{f,j}^k \right)} \quad (10)$$

若所求解 $0 < \tilde{x}_{l,i}^k < C_{l,i}$ ，则最优卸载策略为 $x_{l,i}^{k*} = \tilde{x}_{l,i}^k$ ；若所求解 $\tilde{x}_{l,i}^k < 0$ ，则用户不卸载；若 $C_{l,i} < \tilde{x}_{l,i}^k$ ，则用户全卸载。那么有：给定其他用户的卸载策略，专网用户 i 所求最优卸载策略为

$$x_{l,i}^{k*} = \begin{cases} 0, & \Phi < 0, \\ C_{l,i}, & \Phi > C_{l,i}, \\ \Phi, & \text{其他} \end{cases} \quad (11)$$

式中： $\Phi = \tilde{x}_{l,i}^k$ 。

同样，对于公网用户 j ，求得其最优卸载策略为

$$x_{f,j}^{k*} = \begin{cases} 0, & \Delta < 0, \\ C_{f,j}, & \Delta > C_{f,j}, \\ \Delta, & \text{其他} \end{cases} \quad (12)$$

$$\Delta = \sqrt{\frac{\beta_{f,j} \left(\sum_{i=1}^N x_{l,i}^k + \sum_{m \neq j}^M x_{f,m}^k \right)}{p_{f,j}/r_{f,j}^k - v_{f,j}} R - \left(\sum_{i=1}^N x_{l,i}^k + \sum_{m \neq j}^M x_{f,m}^k \right)} \quad (13)$$

据此，得出结论，无论是专网用户还是公网用户，当已知其他用户的卸载策略时，总能求得其最优卸载策略。

2.2 信息共享下最优卸载策略求解

根据式(11)、(12)，可以得到专网用户和公网用户的最优卸载策略的复合表达式。由式(11)、(12)得，在 $C_{f,j} > \Delta > 0$ 情况下有：

$$x_{\text{total}}^k = \sqrt{\frac{\beta_{f,j} (x_{\text{total}}^k - x_{f,j}^{k*})}{p_{f,j}/r_{f,j}^k - v_{f,j}} R} \quad (14)$$

式中： $x_{\text{total}}^k = \sum_{i=1}^N x_{l,i}^k + \sum_{j=1}^M x_{f,j}^k$ 表示卸载的总数据量。

由式(14)可得

$$x_{f,j}^{k*} = x_{\text{total}}^k - (x_{\text{total}}^k)^2 \frac{p_{f,j}/r_{f,j}^k - v_{f,j}}{R\beta_{f,j}} \quad (15)$$

累加所有公网用户得

$$\sum_{j=1}^M x_{f,j}^k = M x_{\text{total}}^k - (x_{\text{total}}^k)^2 \sum_{j=1}^M \frac{p_{f,j}/r_{f,j}^k - v_{f,j}}{R\beta_{f,j}} \quad (16)$$

同理，对于专网用户，根据式(10)可得

$$x_{l,i}^{k*} = \sum_{i=1}^N x_{l,i}^k + \frac{(\alpha_{l,i} - \beta_{l,i}) \sum_{j=1}^M x_{f,j}^k}{\alpha_{l,i}} - (x_{\text{total}}^k)^2 \frac{p_{l,i}/r_{l,i}^k - v_{l,i}}{\alpha_{l,i} R} \quad (17)$$

累加所有专网用户得

$$\sum_{i=1}^N x_{l,i}^k = N \sum_{i=1}^N x_{l,i}^k + \sum_{j=1}^M x_{f,j}^k \sum_{i=1}^N \frac{(\alpha_{l,i} - \beta_{l,i})}{\alpha_{l,i}} - (x_{\text{total}}^k)^2 \sum_{i=1}^N \frac{p_{l,i}/r_{l,i}^k - v_{l,i}}{\alpha_{l,i} R} \quad (18)$$

由 $x_{\text{total}}^k = \sum_{i=1}^N x_{l,i}^k + \sum_{j=1}^M x_{f,j}^k$ ，联立式(16)、(18)得二

元二次方程，求得其解为 $\sum_{i=1}^N x_{l,i}^k = \Psi_l$ ， $\sum_{j=1}^M x_{f,j}^k = \Psi_f$ ，

并将其代入式(15)、(17)，可计算出 $x_{f,j}^{k*}$ 、 $x_{l,i}^{k*}$ 。

具体算法思路如下。

算法 1 完全信息博弈情况下纳什均衡求解算法

输入：边缘专网用户 $N = \{1, 2, \dots, N\}$ ，公网用户 $M = \{1, 2, \dots, M\}$

输出：每个时隙最优卸载向量

$$\mathbf{x}_l^{k*} = \{x_{l,1}^{k*}, x_{l,2}^{k*}, \dots, x_{l,i}^{k*}, \dots, x_{l,N}^{k*}\}$$

$$\mathbf{x}_f^{k*} = \{x_{f,1}^{k*}, x_{f,2}^{k*}, \dots, x_{f,j}^{k*}, \dots, x_{f,M}^{k*}\}$$

初始化用户自身信息 $p_{l,i}$ 、 $p_{f,j}$ 、 $v_{l,i}$ 、 $v_{f,j}$ 、 $r_{l,i}^k$ 、 $r_{f,j}^k$

所有用户的信息通知全部用户

for 对于时隙 $k = 1, 2, \dots, K$:

根据式(16)、(18)求得 $\sum_{i=1}^N x_{l,i}^k$ 和 $\sum_{j=1}^M x_{f,j}^k$

for 对于专网用户 $i = 1, 2, \dots, N$:

根据式(17)计算最优卸载策略 $x_{l,i}^{k*}$

end for

for 对于公网用户 $j=1, 2, \dots, M$:

根据式(15)计算最优卸载策略 $x_{f,j}^{k*}$

end for

end for

3 不完全信息博弈下卸载策略求解

当用户的信息是不共享的情况下, 用户所获取的信息只有用户自身的系数信息及卸载策略。在连续的时间域中, 对独立用户来说, 这是一个部分可观测的马尔可夫决策过程。

3.1 POMDP 构造

针对任一用户, 该用户在连续时间域中, 需要对任务卸载量做出决策, 每个时隙逐步做出决策的过程可以建模成 POMDP, 记为 $M = \langle S, A, P, R, O, T \rangle$, 式中元素从左到右分别表示问题的状态空间、动作空间、状态转移矩阵、奖励空间、观测空间、状态-观测变换函数。在 POMDP 中, 状态空间 S 是部分已知的, 决策会根据观测空间 O 给出。对于任一专网用户, 它可以观测到过去的卸载策略 $\left\{ \sum_{j=1}^M x_{f,j}^{k-kD}, x_1^{k-kD}, x_1^{k-kD+1}, \dots, x_1^{k-1} \right\}$ 以及用户自身的系数信息。对于任一公网用户来说, 公网用户可以观测到过去的卸载策略 $\{x_f^{k-kD}, x_f^{k-kD+1}, \dots, x_f^{k-1}\}$ 以及用户自身的系数信息, 其中 “\” 表示取余运算。

观测空间: 观测空间 $O = \{O_{1,i}, O_{f,j}\}_{i \in N, j \in M}$, 其中专网用户观测空间为 $O_{1,i} = \{o_{1,i}^k | k \in K\}$, $K = \{1, 2, \dots, K\}$, $o_{1,i}^k = \left\{ \sum_{j=1}^M x_{f,j}^{k-kD}, x_1^{k-kD}, r_{1,i}^{k-kD+1}, x_{1,i}^{k-kD+1}, r_{1,i}^{k-kD+2}, \dots, x_{1,i}^{k-1}, r_{1,i}^k \right\}$, 公网用户观测空间为 $O_{f,j} = \{o_{f,j}^k | k \in K\}$, $o_{f,j}^k = \{x_{f,j}^{k-kD}, r_{f,j}^{k-kD+1}, x_{f,j}^{k-kD+1}, r_{f,j}^{k-kD+2}, \dots, x_{f,j}^{k-1}, r_{f,j}^k\}$ 。

动作空间: 对于专网用户动作空间为 $A_1 =$

$\{A_{1,i}\}_{i \in N}$, 其中 $A_{1,i} = \{x_{1,i}^k | k \in K\}_{i \in N}$, 对于公网用户动作空间为 $A_f = \{A_{f,j}\}_{j \in M}$, 其中 $A_{f,j} = \{x_{f,j}^k | k \in K\}_{j \in M}$ 。

奖励函数: 每个用户都有独立的奖励函数, 以专网用户为例其, 奖励空间为 $R_1 = \{R_{1,i}\}_{i \in N}$, 其中 $R_{1,i} = \{u_{1,i} | k \in K\}_{i \in N}$, 即用户奖励函数等价于效用函数。

3.2 POMDP 过程

将卸载调度模块的策略定义为一个条件概率函数, 表示为 π_θ , 定义 $\pi_\theta: O \times A \rightarrow [0, 1]$, 其中 θ 为用户网络参数。

在 POMDP 中, 以专网用户为例, 状态价值函数为 $V_{\pi_\theta}(o_{1,i}, x_{1,i}) = E \left[\sum_{d=k-kD}^k \gamma^{k-d} u_{1,i}^d | o_{1,i} = o_{1,i}^k \right]$, 动作-状态价值函数为 $Q_{\pi_\theta}(o_{1,i}, x_{1,i}) = E \left[\sum_{d=k-kD}^k \gamma^{k-d} u_{1,i}^d | o_{1,i} = o_{1,i}^k, x_{1,i} = x_{1,i}^k \right]$, 其中 $\gamma \in [0, 1]$ 为折扣因子, 构造目标函数 $L_{1,i} = E \left[\sum_{d=k-kD}^k \gamma^{k-d} u_{1,i}^d \right]$, 根据 Policy Gradient, 有:

$$\nabla_\theta L_{1,i} = E_{\pi_\theta} [\nabla_\theta \log \pi_\theta(x_{1,i} | o_{1,i}) Q_{\pi_\theta}(o_{1,i}, x_{1,i})] = E_{\pi_\theta} [\nabla_\theta \log \pi_\theta(x_{1,i} | o_{1,i}) A_{\pi_\theta}(o_{1,i}, x_{1,i})] \quad (19)$$

其中, $A_{\pi_\theta}(o_{1,i}, x_{1,i}) = Q_{\pi_\theta}(o_{1,i}, x_{1,i}) - b$, $b = V(o_{1,i})$ 。

采用近端策略优化算法 (proximal policy optimization, PPO), 进行重要性采样有

$$\nabla_\theta L_{1,i} \approx E_{\pi_\theta} [\nabla_\theta \log \pi_\theta(o_{1,i}, x_{1,i}) J_{\text{ppo}}^{\theta'}(o_{1,i}, x_{1,i})] \quad (20)$$

其中:

$$J_{\text{ppo}}^{\theta'}(o_{1,i}, x_{1,i}) = J^{\theta'} - \beta KL(\theta, \theta') = \frac{\pi_\theta(x_{1,i} | o_{1,i})}{\pi_{\theta'}(x_{1,i} | o_{1,i})} A_{\pi_\theta}(o_{1,i}, x_{1,i}) - \beta KL(\theta, \theta') \quad (21)$$

因此对于 Actor 网络参数有

$$\theta \leftarrow \theta + \tau \nabla_\theta \bar{L}_{1,i} \quad (22)$$

对于 Critic 网络有

$$\omega \leftarrow \omega - \lambda \nabla_\omega \bar{C}_{1,i} \quad (23)$$

其中: τ 、 λ 分别为 Actor 网络和 Critic 网络的学习率。

$$\nabla_{\theta} \bar{L}_{1,i} = \frac{1}{D} \sum_{k=0}^{D-1} \nabla_{\theta} \log \pi_{\theta}(\mathbf{o}_{1,i}^k, x_{1,i}^k) J_{\text{ppo}}^{\theta}(\mathbf{o}_{1,i}^k, x_{1,i}^k) \quad (24)$$

$$\nabla_{\omega} \bar{H}_{1,i} = \frac{1}{D} \sum_{k=0}^{D-1} [V_{\omega}(\mathbf{o}_{1,i}^k) - Y_{1,i}^k] \frac{dV_{\omega}(\mathbf{o}_{1,i}^k)}{d\omega} \quad (25)$$

$$Y_{1,i}^k = \sum_{d=k-kD}^k \gamma^{k-d} u_{1,i}^d + \gamma^{D-kD} (V_{\omega}(\mathbf{o}_{1,i}^{k-kD+D}) - \sum_{d=k-kD}^{k-kD+D} \gamma^{k-d} u_{1,i}^d) \quad (26)$$

3.3 算法模型

如图 2 所示，用户分为 2 种类型的用户。对于专网用户，用户 i 在每个时隙 k 获取观测值 $\mathbf{o}_{1,i}^k$ ，并根据当前状态通过用户内部的控制器计算其最优的卸载策略 $x_{1,i}^k$ ，同时，根据效用函数 $u_{1,i}$ 计算其回报值，控制器根据回报值进一步调整控制器内部的网络参数。对于公网用户，当专网用户策略选择完毕后，公网用户 j 根据自身获得信息以及当前时隙专网用户的卸载策略，通过自己的控制器获得最优卸载策略 $x_{f,j}^k$ 。此为一个时隙的过程。

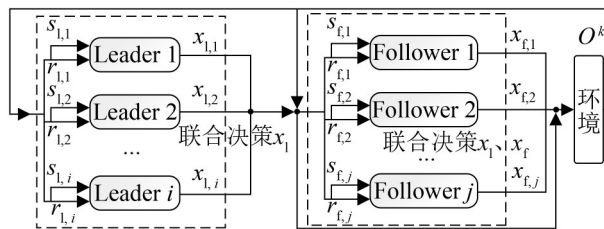


图 2 两阶段优化模型

Fig. 2 Two-stage optimization model

每个用户都拥有自己的一个控制器，图 3 为其控制器逻辑。以专网用户为例，用户在每个时隙将自己观测到的数据、该回合的卸载策略存储到数据存储单元中，Critic 网络根据回报值 $u_{1,i}$ 调整网络参数 $\omega_{1,i}^k$ ，Actor 网络根据观测值以及 Critic 的值函数 $V_{\omega}(\mathbf{o}_{1,i}^k)$ 通过策略梯度优化算法(policy gradient, PG)调整网络参数 $\theta_{1,i}^k$ ，最后输出当前优化策略 $x_{1,i}^k$ 。

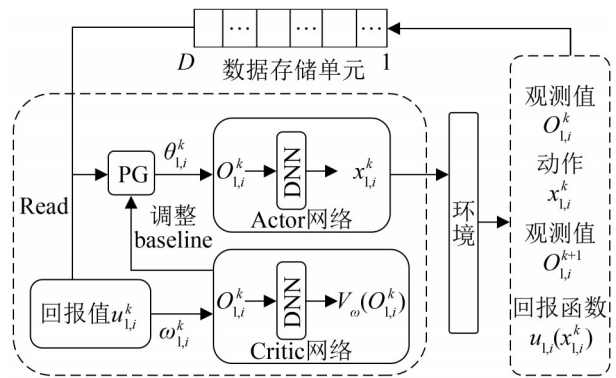


图 3 用户的控制器逻辑

Fig. 3 Controller logic of users

具体算法思路如下：

算法 2 不完全信息博弈情况 TSDRL 算法

输入：边缘专网用户 $N=\{1, 2, \dots, N\}$ ，公网用户 $M=\{1, 2, \dots, M\}$

输出：每个时隙最优卸载向量

$$\mathbf{x}_1^{k*} = \{x_{1,1}^{k*}, x_{1,2}^{k*}, \dots, x_{1,i}^{k*}, \dots, x_{1,N}^{k*}\},$$

$$\mathbf{x}_f^{k*} = \{x_{f,1}^{k*}, x_{f,2}^{k*}, \dots, x_{f,j}^{k*}, \dots, x_{f,M}^{k*}\}$$

初始化用户的系数 $\omega_{1,i}, \theta_{1,i}, \omega_{f,j}, \theta_{f,j}$

初始化经验池容量 D

初始化 $\mathbf{o}_{1,i}^0, \mathbf{o}_{f,j}^0$ 为初始观测状态

for 对于时隙 $k = 1, 2, \dots, K$:

for 对于专网用户 $i = 1, 2, \dots, N$:

观测用户的传输速率 $r_{1,i}^k$ ，更新观测信息

$$\mathbf{o}_{1,i}^{k-1} \rightarrow \mathbf{o}_{1,i}^k$$

存储用户信息 $\{\mathbf{o}_{1,i}^{k-1}, x_{1,i}^{k-1}, \mathbf{o}_{1,i}^k, r_{1,i}^{k-1}\}$ 到 $D_{1,i}$

将 $\mathbf{o}_{1,i}^k$ 输入 Actor 网络并输出 $x_{1,i}^{k*}$

$$\text{计算 } \sum_{j=0}^M x_{f,j}^k = \sum_{j=0}^M x_{f,j}^{k-k\%D}$$

由式(2)，计算用户的回报值 $u_{1,i}(x_{1,i}^{k*})$

end for

for 对于公网用户 $j = 1, 2, \dots, M$:

观测用户的传输速率 $r_{f,j}^k$ ，更新观测信息

$$\mathbf{o}_{f,j}^{k-1} \rightarrow \mathbf{o}_{f,j}^k$$

存储用户信息 $\{\mathbf{o}_{f,j}^{k-1}, x_{f,j}^{k-1}, \mathbf{o}_{f,j}^k, r_{f,j}^{k-1}\}$ 到 $D_{f,j}$

将 $\mathbf{o}_{f,j}^k$ 输入 Actor 网络并输出 $x_{f,j}^{k*}$

由式(4)，计算回报值 $u_{f,j}(x_{f,j}^{k*})$

end for

```

if  $k \% D = 0$ :
    for 对于专网用户  $i = 1, 2, \dots, N$ :
        根据式(22)、(23)更新  $\theta_{1,i}^k, \omega_{1,i}^k$ 
    end for
    for 对于公网用户  $j = 1, 2, \dots, M$ :
        根据式(22)、(23)更新  $\theta_{f,j}^k, \omega_{f,j}^k$ 
    end for
    清空数据存储单元  $D$ 
end for
    
```

对于 TSDRL 算法, 其 Actor 网络包含 2 层节点数为 100 和 50 的全连接层, Critic 网络同样包含 2 层节点数为 100 和 50 的全连接层。神经网络的复杂度为 $O\left(\sum_{l=1}^L s_l \cdot s_{l-1}\right)$, 其中, s_l 表示第 l 层的节点数^[22]。在算法 2 中, 每个用户循环更新, 因此仿真算法的时间复杂度为

$$O(K(M+N) \sum_{l=1}^L s_l \cdot s_{l-1})$$

在现实中, 每个用户是并行运算的, 其时间复杂度为

$$O(K \sum_{l=1}^L s_l \cdot s_{l-1})$$

算法流程如图 4 所示。

4 实验结果与分析

4.1 参数设置

使用 Python3.7 和 Tensorflow1.1 进行仿真实验。

在仿真中设置 MEC 服务器 1 个, 设置 3 个专网用户和 5 个公网用户, 本节参数设定如表 1 所示。

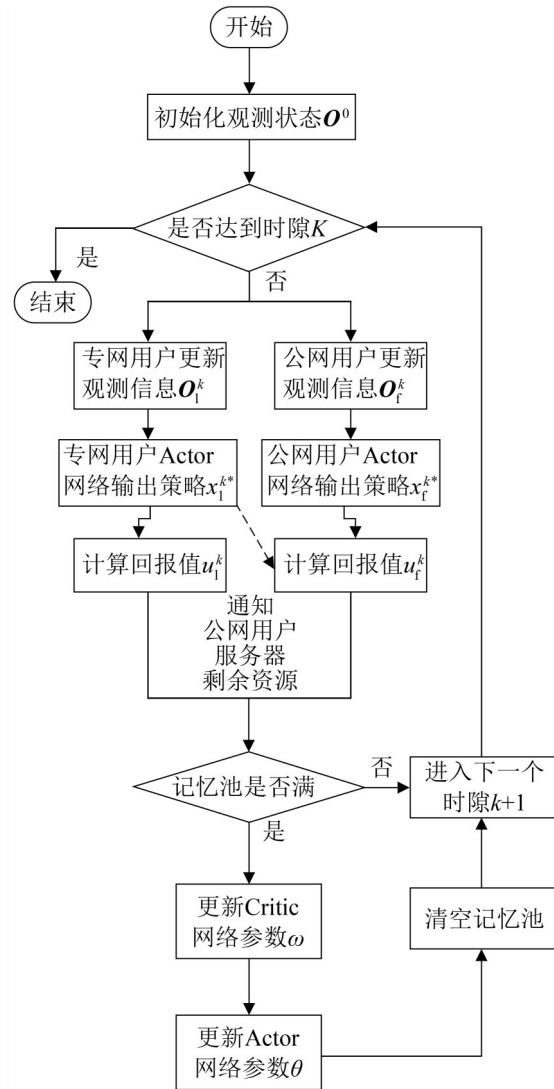


图 4 TSDRL 算法流程

Fig. 4 TSDRL algorithm flow chart

表 1 实验设置参数

Table 1 Experimental parameter setting

参数	值	参数	值
计算总资源 R (Mbps)	50	专网用户 2 传输速率 $r_{1,2}^k$ (Mb/s)	1/6
专网用户任务总量 $C_{1,i}$ (Mb)	10	专网用户 3 传输速率 $r_{1,3}^k$ (Mb/s)	1/5.6
公网用户任务总量 $C_{f,j}$ (Mb)	1	公网用户 1 传输速率 $r_{f,1}^k$ (Mb/s)	1/8.4
单位传输时间成本 ρ (J/s)	1	公网用户 2 传输速率 $r_{f,2}^k$ (Mb/s)	1/8.3
单位数据消耗功率成本 v (J/Mb)	0.1	公网用户 3 传输速率 $r_{f,3}^k$ (Mb/s)	1/8.2
专网用户本地计算资源 $F_{1,i}^k$ (Mb/s)	0.1	公网用户 4 传输速率 $r_{f,4}^k$ (Mb/s)	1/8.1
公网用户本地计算资源 $F_{f,j}^k$ (Mb/s)	0.01	公网用户 5 传输速率 $r_{f,5}^k$ (Mb/s)	1/8
专网用户 1 传输速率 $r_{1,1}^k$ (Mb/s)	1/6.4		

在仿真前, 求解完全信息博弈下的解。通过表 1 的参数, 初始化所有用户的自身参数, 根据式(16)、(18)求得 $\sum_{i=1}^N x_{l,i}^k = 3.45$ 和 $\sum_{j=1}^M x_{f,j}^k = 2.21$ 。对所有专网用户依次使用式(17)求得最优卸载策略得 $\mathbf{x}_l^{k*} = \{x_{l,1}^{k*} = 0.86, x_{l,2}^{k*} = 1.15, x_{l,3}^{k*} = 1.43\}$, 对所有公网用户依次使用式(15)得 $\mathbf{x}_f^{k*} = \{x_{f,1}^{k*} = 0.32, x_{f,2}^{k*} = 0.38, x_{f,3}^{k*} = 0.44, x_{f,4}^{k*} = 0.51, x_{f,5}^{k*} = 0.57\}$, \mathbf{x}_l^{k*} 、 \mathbf{x}_f^{k*} 为最优卸载策略。

对于不完全信息博弈, 参照图 4 的流程图。用户分为 3 个专网用户和 5 个公网用户, $k=1$ 时先更新观测状态: $\mathbf{o}_{l,1}^1 = \{x_{l,1}^0, r_{l,1}^1 = 1/6.4\}, \dots, \mathbf{o}_{f,5}^1 = \{x_{f,5}^0, r_{f,5}^1 = 1/8\}$, 其中 $x_{l,1}^0, x_{l,2}^0, x_{l,3}^0, x_{f,1}^0, \dots, x_{f,5}^0$ 为随机值, 初始化网络参数 $\omega_{l,1}, \dots, \omega_{f,5}, \theta_{l,1}, \dots, \theta_{f,5}$ 为正态分布的随机数。Actor 网络根据观察状态输出策略 $x_{l,1}^{1*}, x_{l,2}^{1*}, x_{l,3}^{1*}, x_{f,1}^{1*}, \dots, x_{f,5}^{1*}$, 专网用户假定公网用户的策略为随机策略并计算回报值 $u_{l,1}^{1*}, u_{l,2}^{1*}, u_{l,3}^{1*}$, 公网用户获取专网用户竞争后的服务器剩余资源信息并计算回报值 $u_{f,1}^{1*}, \dots, u_{f,5}^{1*}$, 同时将信息 $\{r_{l,1}^1, x_{l,1}^{1*}\}, \dots, \{r_{f,5}^1, x_{f,5}^{1*}\}$ 存储到存储单元 D 。 $k=2$ 时, 更新观测状态 $\mathbf{o}_{l,1}^2, \dots, \mathbf{o}_{f,5}^2$, 以此类推。当记忆存储单元满时(即 $kD=0$), 根据式(22)、(23)更新 ω 、 θ 的值, 并计算 $\sum_{j=0}^M x_{f,j}^{k-D}$ 保存在专网用户的记忆存储单元中。

4.2 实验结果分析

4.2.1 算法收敛性分析

如图 5 为 TSDRL 算法的收敛性验证。可以观察到, 在不完全信息博弈的情况下, 该算法能使用户的卸载策略及效用均收敛至完全信息博弈下的纳什均衡解(见图 NE 所示曲线)。此外可以观测到专网用户要比公网用户更快收敛。TSDRL 算法是主从依次执行的算法, 能优先考虑专网用户, 专网用户的数据量大, 先使专网用户策略趋近于最优策略, 然后再逐步调整公

网用户的策略。

4.2.2 改变信道传输速率系统变化分析

如图 6 所示, 在改变专网用户 2 的传输速率时, 用户的效用值、卸载值、时延、能耗跟完全博弈情况(NE)下接近。

图 6(a)(b)为在改变专网用户 2 的传输速率时, 用户的效用值变化。当专网用户 2 的传输速率从 1/6.8 提高到 1/5.2 时, 由于其传输时延更小, 卸载更多数据能获取更大收益, 因此其效用值升高。同时, 由于专网用户 2 获取的服务器资源增加, 其他用户能竞争到的服务器计算资源下降, 因此其他用户的效用值都有一定程度的下降。

图 6(c)为在改变专网用户 2 的传输速率时, 用户的计算卸载策略的变化。可以看到, 当专网用户 2 的传输速率上升, 对其他专网用户的计算卸载策略影响较小, 而对于其他公网用户都有比较明显的将数据量放在本地计算的倾向。这是因为在 Stackelberg 博弈中, 专网用户具有博弈优势, 服务器资源会优先提供给专网用户, 从而保障专网用户性能稳定。

图 6(d)为在改变专网用户 2 的传输速率时, 用户的时延的变化。可以看到, 提高专网用户 2 的传输速率该用户时延下降。同时, 公网用户 5 时延变化趋势是先下降再上升, 这是因为用户时延等于计算卸载时延和本地计算时延中的最大值, 提高专网用户的传输速率, 公网用户 5 缺少竞争力, 卸载量减少, 计算卸载时延减少, 但当卸载量减少到一定值时, 本地计算时延大于计算卸载时延, 此时用户时延升高。

如图 6(e)为用户的能耗, 当提高专网用户 2 的传输速率, 可以看到专网用户 2 的能耗先上升再下降, 用户能耗主要由卸载能耗和本地运行能耗组成, 当卸载量增加时卸载能耗上升, 同时本地运行能耗下降。

4.2.3 算法性能分析

图 7 为在 TSDRL、D-DRL^[14]和随机算法 3 种算法下, 计算所有用户的总时延、总能耗、最大时延, 以及在不同的公网用户计算力条件下的最大时延, 其中网络参数和权重系数都保持一致。图 7(a) 所示, TSDRL 算法比随机算法所消耗的总时延减少了 54.8%, 比 D-DRL 算法消耗的总时延减少了 20.81%。图 7(b) 所示, TSDRL 比 D-DRL 算法降低 3.38% 能耗。这是因为针对混合专网结构, 两阶段优化模型根据专网用户和公网用户特性制定不同的效用函数, 即使在信道时变的环境下, 当系统达到纳什均衡时, 也能使系统整体的时延和能耗降低。图 7(c)、(d) 所示, 比较各算法最大时延。所谓最大时延指当某个用户比所有用户处理的时延都要大, 则该时延为最大时延。该变量主要考虑, MEC 服务

器在并行处理卸载任务的速度。由图 7(c) 可以看到改变专网用户 2 传输速率, TSDRL 算法最大时延总是最小。现考虑, 在公网用户的本地计算资源有限的情况下, 各个算法的性能。如图 7(d) 所示, 当逐渐降低公网用户的本地计算力时, 可以看到 D-DRL 算法的最大时延增加, 而 TSDRL 算法的最大时延也上升, 但是上升趋势相对较缓。当公网用户的计算能力仅剩 0.02 Mb/s 时, 采用 D-DRL 的公网用户最大时延达到了 49.9, 比随机算法的时延还高。这主要是因为 D-DRL 算法中, 当专网用户支配了绝大多数的 MEC 服务器资源, 且公网用户的本地资源不足时, 容易引起公网用户的宕机。而二阶段博弈的 TSDRL 算法则考虑到这个问题, 专网用户能考虑到公网用户本地运行资源不足的情况, 为了防止其宕机, 能给公网用户保留一定的 MEC 服务器资源。

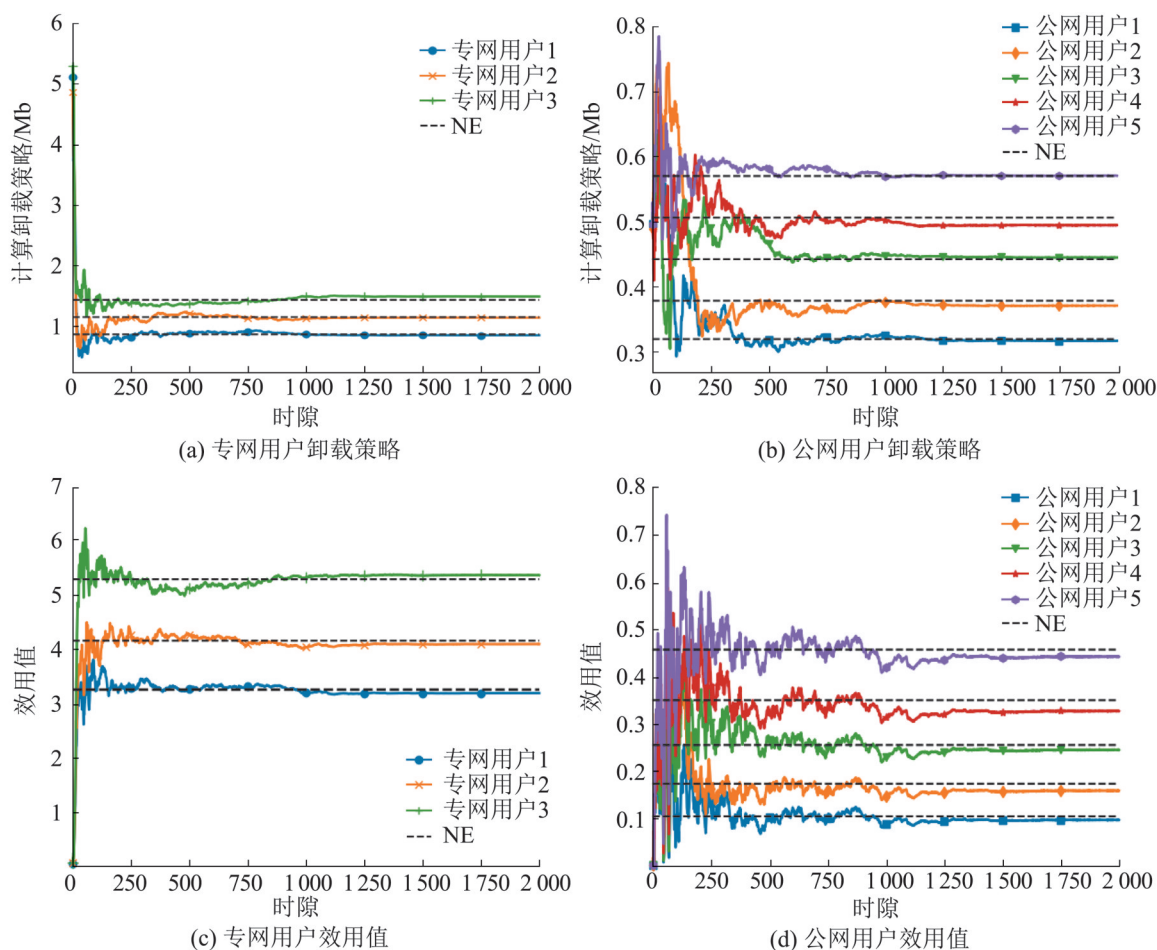


图 5 TSDRL 收敛性

Fig. 5 Convergence of TSDR

<http://www.china-simulation.com>

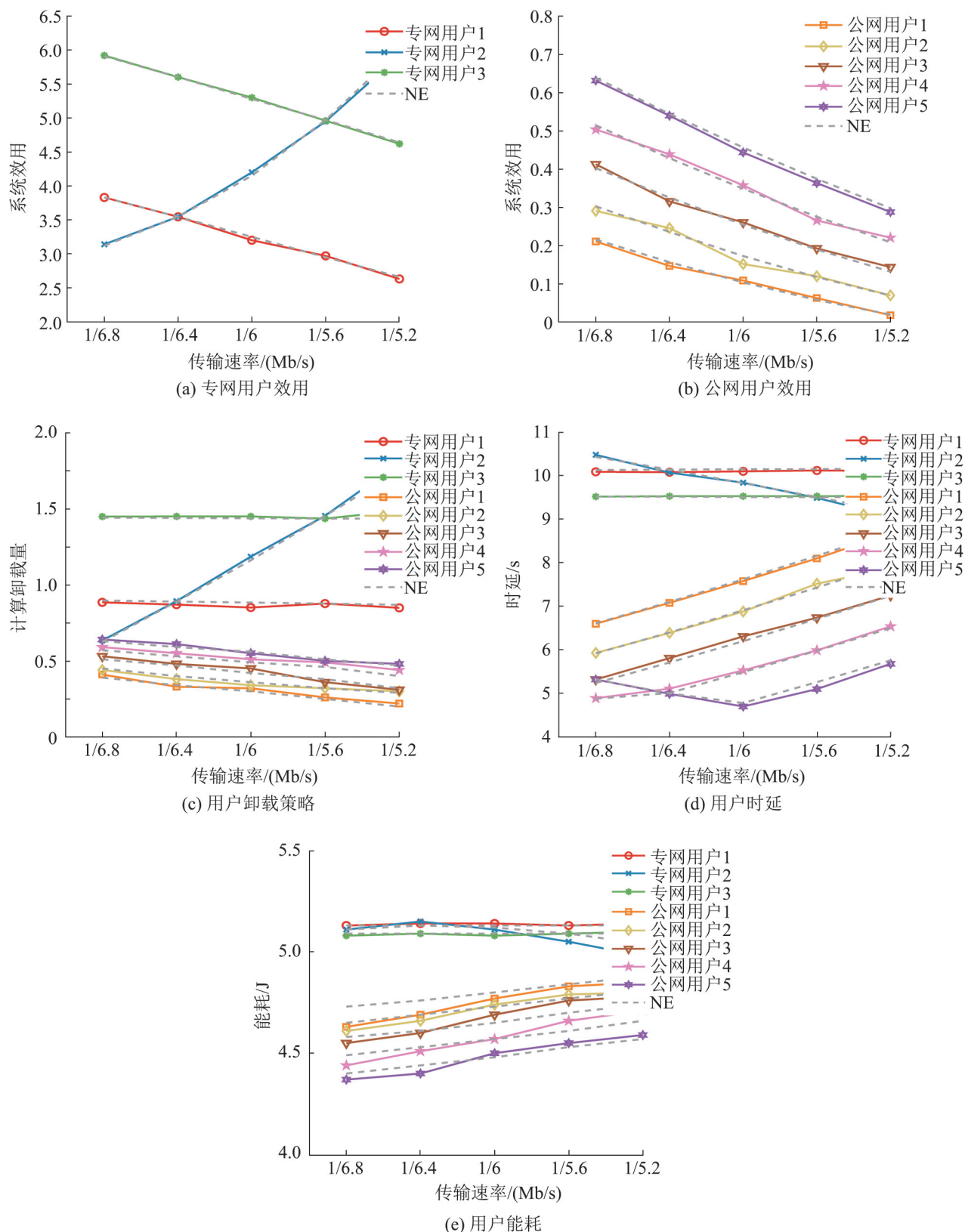


图 6 改变专网用户 2 信道时系统分析

Fig. 6 System analysis when leader 2 channel changed

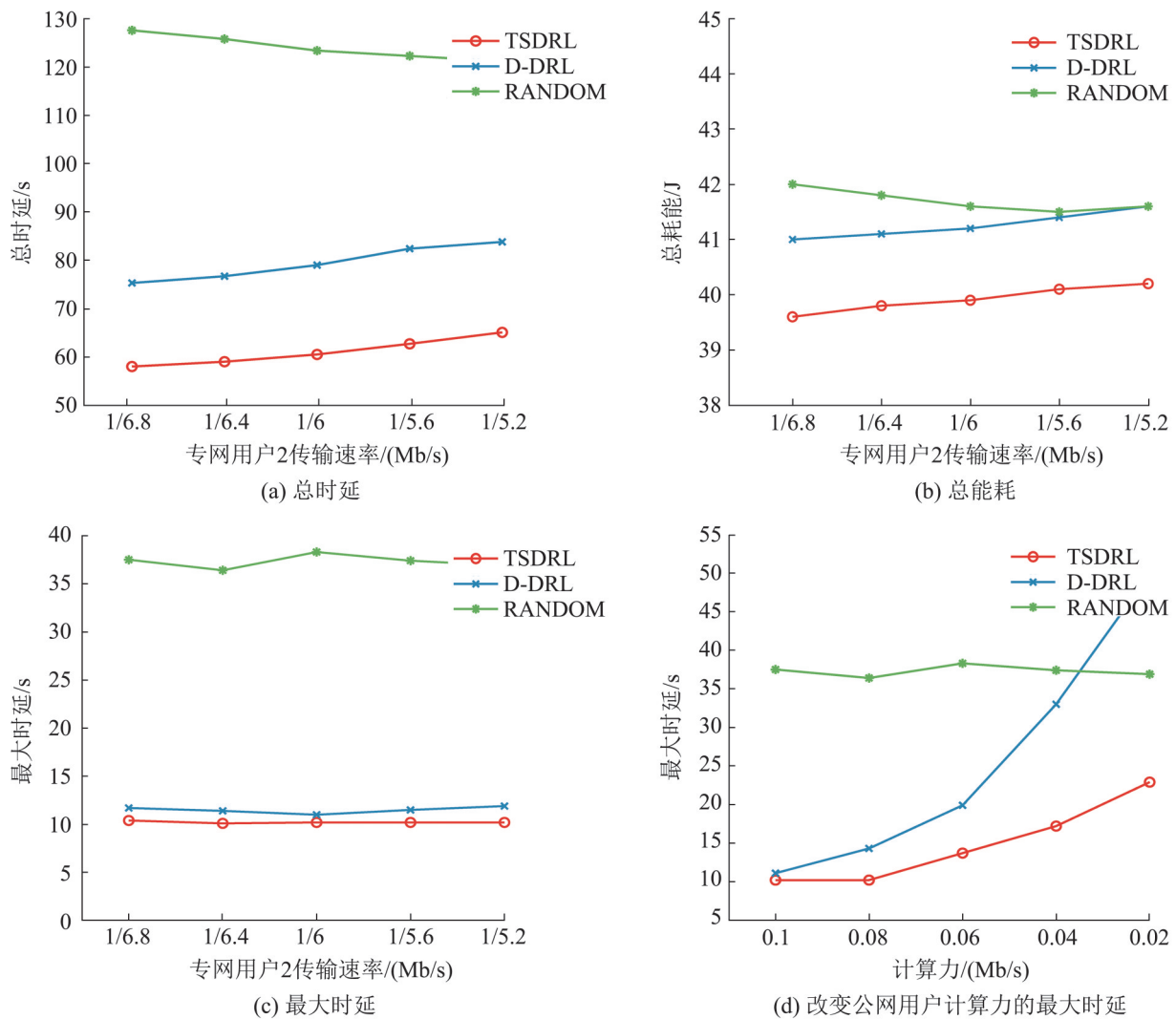


图7 3种不同算法性能比较

Fig. 7 Performance comparison of 3 different algorithms

5 结论

本文研究了边缘计算环境中动态资源分配的问题, 主要贡献如下:

(1) 针对5G混合专网模式, 构建了网络的系统模型。将以往研究中只有单一的卸载用户分为面向行业的专网用户及面向公众的公网用户, 并将两类用户竞争服务器资源问题建模成Stackelberg博弈模型。

(2) 本文证明了在完全信息博弈下该模型存在纳什均衡解, 并得出解。通过对二阶段博弈的分解, 证明了博弈的效用函数为凸函数, 并求得两

种用户的卸载量和, 通过凸优化方法求出每位用户的最优策略。

(3) 本文提出一种基于二阶段深度强化学习的最优卸载算法(two-stage DRL, TSDRL), 在不完全信息博弈下, 将问题转化为二阶段深度强化学习, 依次求解该模型中专网用户和普通用户最优卸载策略。

实验仿真结果也表明, 本文提出的算法能有效地优化目标值。然而, 在边缘计算卸载过程中, 是多服务器对多用户的, 随着用户的移动, 整个网络的拓扑结构也会发生变化, 从而影响用户的QoE。因此, 未来考虑拓扑结构变化的多边缘云-

多用户中, 端-边-云协作的计算卸载优化问题。

参考文献:

- [1] Strinati E C, Haustein T, Maman M, et al. Beyond 5G Private Networks: the 5G CONNI Perspective[C]//2020 IEEE Globecom Workshops(GC Wkshps. Taiwan, China: IEEE, 2020: 1-6.
- [2] Li X, Garcia-Saavedra A, Costa-Perez X, et al. 5Growth: an End-to-End Service Platform for Automated Deployment and Management of Vertical Services over 5G Networks[J]. IEEE Communications Magazine (S0163-6804), 2021, 59(3): 84-90.
- [3] Xu J, Chen L, Zhou P. Joint Service Caching and Task Offloading for Mobile Edge Computing in Dense Networks[C]//IEEE INFOCOM 2018-IEEE Conference on Computer Communications. Honolulu, HI, USA: IEEE, 2018: 207-215.
- [4] 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 Unloading Strategy in Mobile Edge Computing[J]. Computer Science, 2019, 46(11): 32-40.
- [5] Walia J S, Hämmäinen H, Flinck H. Future Scenarios and Value Network Configurations for Industrial 5G[C]//2017 8th International Conference on the Network of the Future(NOF). London, United Kingdom: IEEE, 2017: 79-84.
- [6] Guo H Z, Zhang J, Liu J J, et al. Energy-Efficient Task Offloading and Transmit Power Allocation for Ultra-Dense Edge Computing[C]//2018 IEEE Global Communications Conference(GLOBECOM). United Arab Emirates: IEEE, 2018: 1-6.
- [7] Liu J, Mao Y, Zhang J, et al. Delay-Optimal Computation Task Scheduling for Mobile-Edge Computing Systems [C]//2016 IEEE International Symposium on Information Theory(ISIT). Barcelona, Spain: IEEE, 2016: 1451-1455.
- [8] Kamoun M, Labidi W, Sarkiss M. Joint Resource Allocation and Offloading Strategies in Cloud Enabled Cellular Networks[C]//2015 IEEE International Conference on Communications(ICC). United Kingdom: IEEE, 2015: 5529-5534.
- [9] You C, Huang K, Chae H, 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] Chen M H, Liang B, Dong M. Joint Offloading Decision and Resource Allocation for Multi-User Multi-Task Mobile Cloud[C]//2016 IEEE International Conference on Communications(ICC). Kuala Lumpur, Malaysia: IEEE, 2016: 1-6.
- [11] 李长云, 黎建波, 徐曦, 等. 具有能量收集设备的移动边缘计算资源分配研究[J]. 系统仿真学报, 2022, 34(11): 2313-2322.
Li Changyun, Li Jianbo, Xu Xi, et al. Research on Mobile Edge Computing Resource Allocation with Energy Harvesting Device[J]. Journal of System Simulation, 2022, 34(11): 2313-2322.
- [12] Tran T X, Pompili D. Joint Task Offloading and Resource Allocation for Multi-Server Mobile-Edge Computing Networks[J]. IEEE Transactions on Vehicular Technology(S0018-9545), 2018, 68(1): 856-868.
- [13] Cao H, Cai J. Distributed Multiuser Computation Offloading for Cloudlet-Based Mobile Cloud Computing: A Game-Theoretic Machine Learning Approach[J]. IEEE Transactions on Vehicular Technology(S0018-9545), 2017, 67(1): 752-764.
- [14] Zhan Y, Guo S, Li P, et al. A deep Reinforcement Learning Based Offloading Game in Edge Computing[J]. IEEE Transactions on Computers(S0018-9340), 2020, 69(6): 883-893.
- [15] Asheralieva A, Niyato D. Bayesian Reinforcement Learning and Bayesian Deep Learning for Blockchains with Mobile Edge Computing[J]. IEEE Transactions on Cognitive Communications and Networking(S2332-7731), 2020, 7(1): 319-335.
- [16] Li F, Yao H, Du J, et al. Stackelberg Game-Based Computation Offloading in Social and Cognitive Industrial Internet of Things[J]. IEEE Transactions on Industrial Informatics(S1551-3203), 2019, 16(8): 5444-5455.
- [17] Kotulski Z, Niewolski W, Nowak T W, et al. New Security Architecture of Access Control in 5G MEC[C]//International Symposium on Security in Computing and Communication. Singapore: Springer, 2020: 77-91.
- [18] 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(7): 1468-1481.
Wu Xuewen, Liao Jingxian. Game-based Resource Allocation and Task Offloading Scheme in Collaborative Cloud-Edge Computing System[J]. Journal of System Simulation, 2022, 34(7): 1468-1481.
- [19] Wan X, Sheng G, Li Y, et al. Reinforcement Learning Based Mobile Offloading for Cloud-Based Malware Detection[C]//GLOBECOM 2017-2017 IEEE Global Communications Conference. Singapore: IEEE, 2017: 1-6.
- [20] Bai Y, Chen L, Song L, et al. Risk-Aware Edge Computation Offloading Using Bayesian Stackelberg

- Game[J]. IEEE Transactions on Network and Service Management(S1932-4537), 2020, 17(2): 1000-1012.
- [21] Pang J S, Fukushima M. Quasi-Variational Inequalities, Generalized Nash Equilibria, and Multi-Leader-Follower Games[J]. Computational Management Science (S1619-697X), 2005, 2(1): 21-56.
- [22] He K, Sun J. Convolutional Neural Networks at Constrained Time Cost[C]//IEEE Conference on Computer Vision and Pattern Recognition. United States: IEEE, 2015: 5353-5360.