

文章编号:1007-5321(2021)05-0041-07

DOI:10.13190/j.jbupt.2021-005

MEC系统中面向网络切片的3C联合资源分配算法

郑远鹏¹, 张天魁¹, 朱光宇¹, 沈 鸿²

(1. 北京邮电大学 信息与通信工程学院, 北京 100876; 2. 中国电信股份有限公司 北京分公司, 北京 100010)

摘要: 移动边缘计算(MEC)相关研究已经成为未来移动网络的热点研究之一。在基于网络切片的雾无线接入网络中,提出了一种面向网络切片的 MEC 系统通信计算缓存(3C)联合的资源分配算法。给出了面向网络切片的多 MEC 协作资源分配模型,其中考虑了无线接入带宽与回程链路带宽的限制以及 MEC 系统计算与缓存资源分配对网络切片服务时延的影响;基于服务时延定义了用户获得资源的效用值,构造了系统效用值最大化问题优化不同网络切片下的用户接入、计算与缓存资源分配,并采用连续凸近似交替迭代方法获得近似最优解。对所提算法性能进行了仿真验证,仿真结果表明,所提算法实现了系统总效用值优化,提高了面向网络切片的 MEC 系统资源利用效率。

关键词: 网络切片; 移动边缘计算; 连续凸近似; 资源联合分配

中图分类号: TN929.5

文献标志码: A

Network Slicing-Oriented Joint Allocation Algorithm of 3C Resources in MEC Systems

ZHENG Yuan-peng¹, ZHANG Tian-kui¹, ZHU Guang-yu¹, SHEN Hong²

(1. School of Information and Communication Engineering, Beijing University of Post and Telecommunication, Beijing 100876, China;

2. Beijing Branch, China Telecom Company Limited, Beijing 100010, China)

Abstract: Mobile edge computing (MEC) has become one of the emerging research topics of future mobile network. In the fog radio access network based on network slicing, a network slicing-oriented communication, computation and caching (3C) joint resource allocation algorithm is proposed for MEC systems. A network slicing-oriented resource allocation model with multi-MEC cooperation is given. In this model, the limitations of wireless and backhaul bandwidth and the influence of MEC computation and caching resource allocation on service delay of network slicing are considered. The utility value of resources obtained by the user is defined by the service delay. A system utility maximization problem is proposed, which takes user association in different slices, computation and cache resources allocation as optimization variables. An iterative algorithm based on successive convex approximation is adopted to acquire approximate optimal solution. The performance of the proposed algorithm is verified by simulation. The results show that the proposed algorithm can optimize the total utility value of the system and improve the resource utilization efficiency of the network slicing-oriented MEC system.

Key words: network slicing; mobile edge computing; successive convex approximation; joint allocation of resources

收稿日期: 2021-01-13

基金项目: 国家自然科学基金项目(61971060)

作者简介: 郑远鹏(1998—), 男, 博士生。

通信作者: 张天魁(1980—), 男, 教授, 博士生导师, E-mail: zhangtiankui@bupt.edu.cn.

为了满足未来移动通信网络的海量内容传输与复杂计算需求,移动边缘计算(MEC, mobile edge computing)技术得到了很好地运用^[1]. MEC将计算资源从云端下拉到无线接入网络侧,可以更高效地在本地完成对数据的处理,可以显著提升服务的时延和吞吐量性能,并有效地降低核心网的业务拥塞^[2]. 在MEC系统中,边缘化的计算存储能力会催生更多服务类型和垂直行业应用,因此为保证端到端的可靠性,提高服务质量,网络切片技术成为一种理想的解决方案. 在MEC系统中引入网络切片,可以很好地结合MEC的系统优势和网络切片的技术优势,满足多样化的低时延高可靠的移动服务.

目的,已有研究主要考虑MEC系统资源分配问题. Bi等^[3]研究了单用户单MEC系统下,移动用户计算任务卸载和MEC缓存放置的问题. Wen等^[4]研究了单边缘服务节点、多服务模式与多用户并存的场景,实现了对该单边缘节点系统的缓存、计算和通信资源进行联合分配与优化. Ren等^[5]主要针对边缘计算网络中存在的边缘节点算力不足的情况,研究了多边缘节点之间的协作计算和资源调度问题. 上述研究没有在MEC系统中结合网络切片场景考虑资源分配. 目前有少数文献研究了面向网络切片的MEC系统资源分配问题. Wang等^[6]给出了通信计算缓存资源并存的边缘网络切片的理论性能分析. Rezvani等^[7]研究在多载波非正交频分多址的异构MEC网络中为视频类网络切片提供多比特速率服务,提出了一种传输感知的缓存策略. 目前,针对多MEC协作及满足多种网络切片类型的资源分配研究尚属空白.

受上述启发,提出了一种面向网络切片的MEC系统通信计算缓存资源联合分配算法,主要考虑用户接入和计算与缓存资源分配的联合优化,以提升网络切片服务时延,满足用户差异化应用需求. 主要工作包括:①基于服务时延定义了用户获得资源的效用值,提出了系统总效用值最大化的优化问题;②为了求解所提出的优化问题,提出了一种基于连续凸近似(SCA, successive convex approximation)交替迭代算法求解该优化问题,实现了系统总效用的优化;③对所提算法性能进行了仿真验证.

1 系统模型与问题建模

1.1 网络模型

考虑雾无线接入网络(F-RAN, frog-radio access network)中,在1个宏基站(MBS, micro base station)

覆盖范围内部署 K 个微基站(SBS, small base station). MBS和SBS均部署边缘计算设备,形成MEC系统,为 N 个网络切片提供端到端服务. MBS利用软件定义网络(SDN, software defined network)控制器管理覆盖范围内的 K 个MEC协作的资源分配. 为了简化模型且不失一般性,笔者将时频资源等效为归一化的频率资源,即用户间采用频分多址的方式接入SBS. 假设SBS间频率复用率为1,即SBS间共享系统带宽资源. 基于F-RAN的MEC系统资源包括:通信资源(回程链路带宽、无线接入带宽)、缓存资源和计算资源.

图1所示为基于F-RAN的面向网络切片的多MEC协作资源分配架构. 其中,F-RAN中不同网络切片对应不同应用场景中的业务类型,具有不同的资源需求. 例如,在AR/VR应用中,用户请求的内容数据主要包括可穿戴设备的传感器通用数据和特殊场景的内容渲染数据. 在MEC系统中,通用数据如果已经被缓存,则可以直接发送给用户,无需通过回程链路从云服务器获取,从而可以减少内容传输时延;内容渲染数据可以通过MEC的计算能力为用户提供计算任务卸载,减小用户的计算处理时延. 系统通过多MEC协作的方式并根据网络切片对用户请求进行优化的资源分配. 在相同SDN控制器覆盖范围内的多个MEC协作进行面向网络切片的资源分配.

令网络切片集合为 $N^s = \{1, \dots, n, \dots, N\}$, SBS集合为 $K^s = \{1, \dots, k, \dots, K\}$, 用户集合为 $U^s = \{1, \dots, u, \dots, U\}$, 切片 n 中文件的集合为 $M_n^s = \{1, \dots, m, \dots, M_n\}$. 记第 k 个SBS的无线接入带宽为 W_k 、缓存容量为 C_k 、计算容量为 F_k . 配备MEC的SBS点 k 与 k' 间的数据传输速率表示为 $R_{k,k'}$. 不同网络切片提供不同类型的服务,具有不同的资源需求. 设用户 u 对切片 n 中内容 m 的请求为 $q_{u,n,m} \in \{0, 1\}$. 当用户请求了网络切片 n 中的内容 m ,需要的传输的内容大小为 $C_{n,m}$ (bits),需要计算处理的数据量为 $L_{n,m}$ (bits),相应的所需MEC计算资源为 $f_{n,m}$ (Giga-cycle/s),需要传输的计算处理后数据内容大小为 $O_{n,m}$ (bits).

定义 $x_{n,u,k}$ 为请求了网络切片 n 的用户 u 接入第 k 个SBS的指示变量. $x_{n,u,k} \in \{0, 1\}$, $x_{n,u,k} = 1$ 表示接入,否则 $x_{n,u,k} = 0$. 用户 u 接入第 k 个SBS请求网络切片 n 内容时获得的无线接入带宽为 $w_{n,u,k}$. 采用用户均分的方式分配每个SBS的无线接入带

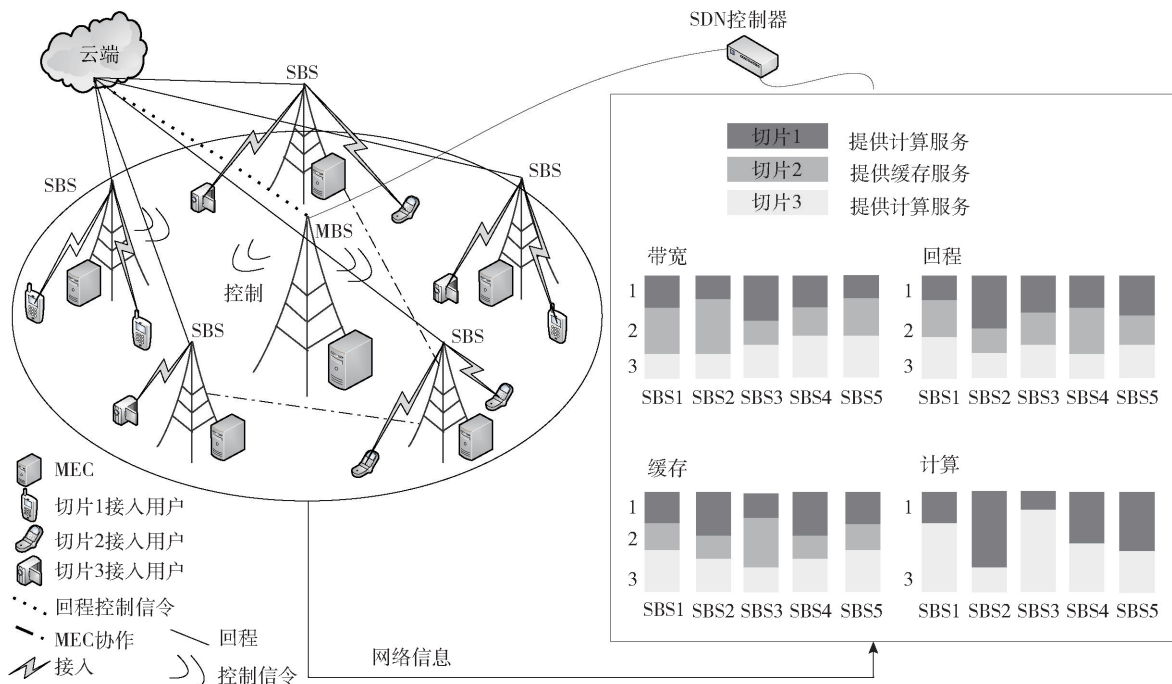


图1 基于 F-RAN 的面向网络切片的多 MEC 协作资源分配架构

宽资源,所以 $w_{n,u,k} = x_{n,u,k} W_k / N_k$,其中 $N_k = \sum_u x_{n,u,k}$ 为接入第 k 个 SBS 的用户总数。根据香农容量公式,请求网络切片 n 的用户 u 接入到第 k 个 SBS 时获得的无线链路传输速率 $r_{n,u,k}$ 为

$$r_{n,u,k} = w_{n,u,k} \log \left(1 + \frac{g_{u,k} P_k}{I_{u,k} + \sigma^2} \right) \quad (1)$$

其中: $g_{u,k}$ 为第 k 个 SBS 与用户 u 之间的无线信道增益,包括大尺度衰落、阴影衰落和小尺度衰落; P_k 为第 k 个 SBS 的发送功率, $I_{u,k}$ 为受到的同频干扰,且 $I_{u,k} = \sum_{i \in K, i \neq k} g_{u,i} P_i$; σ^2 为加性高斯白噪声 (AWGN, additive white Gaussian noise) 的功率谱密度。当用户 u 请求网络切片 n 的内容 m 时,内容 m 从用户 u 到第 k 个 SBS 的无线链路中的传输时延为

$$t_{u,n,m,k}^T = \frac{C_{n,m} q_{u,n,m}}{r_{n,u,k}} \quad (2)$$

内容 m 在第 k 个 SBS 的 MEC 服务器上的数据处理时延为

$$t_{u,n,m,k}^C = \frac{L_{n,m} q_{u,n,m}}{f_{n,m}} + \frac{O_{n,m} q_{u,n,m}}{r_{n,u,k}} \quad (3)$$

内容 m 在第 k 和第 k' 个 SBS 间的传输时延为

$$t_{u,n,m}^M = \frac{C_{n,m} q_{u,n,m}}{R_{k,k'}} \quad (4)$$

用户 u 从云端服务器获取内容 m 的传输时延为

$$t_{u,n,m}^B = \frac{C_{n,m} q_{u,n,m}}{R^B} \quad (5)$$

其中 R^B 为回程链路容量。

假设云服务器具有充足的存储空间和计算能力,因此忽略云服务器上对渲染数据的计算时延。同时考虑到相比于下行传输的内容数据量,用户发送的上行数据量非常小,可以忽略上行传输时延。

由于用户请求既存在内容下载的要求,又有计算卸载的要求,故将用户请求看作缓存资源请求和计算资源请求两部分。

设 $y_{u,n,m,k'} \in \{0,1\}$,当用户 u 接入第 k 个 SBS 请求网络切片 n 中的内容 m 被缓存在第 k' 个配备 MEC 的 SBS 时, $y_{u,n,m,k'} = 1$,即用户可以在 MEC 中直接取得缓存的内容;否则 $y_{u,n,m,k'} = 0$ 。MEC 间协作场景中,任意 2 个 SBS 间相互连通,无需考虑 SBS 之间进行内容传输的多跳时延。用户 u 请求网络切片 n 中的内容 m 的缓存服务时延为

$$T_{u,n,m,k}^{\text{cac}} = y_{u,n,m,k'} (t_{u,n,m,k}^T + t_{u,n,m}^M) + (1 - y_{u,n,m,k'}) t_{u,n,m}^B \quad (6)$$

当内容 m 缓存在用户 u 所接入的第 k 个 SBS 上,相当于 $R_{k,k'} \rightarrow \infty$,此时 $t_{u,n,m}^M = 0$ 。

当用户 u 通过第 k' 个 SBS 请求网络切片 n 的内容 m 需要 MEC 计算资源时,设 $z_{u,n,m,k'} \in \{0,1\}$ 用于表示用户 u 是否需要在第 k' 个 SBS 进行计算任务卸载, $z_{u,n,m,k'} = 1$ 表示用户 u 在第 k' 个 SBS 的 MEC 上获得计算资源;否则 $z_{u,n,m,k'} = 0$ 。则用户 u 请求网

络切片 n 中内容 m 的计算服务时延表示为

$$T_{u,n,m,k}^{\text{com}} = z_{u,n,m,k'} \left(\frac{O_{n,m} q_{n,m}}{R_{k,k'}} + t_{u,n,m,k}^{\text{C}} \right) + (1 - z_{u,n,m,k'}) t_{u,n,m}^{\text{B}} \quad (7)$$

在式(7)中,当内容 m 缓存在用户 u 所接入的第 k 个 SBS 时,相当于 $R_{k,k'} \rightarrow \infty$, 此时 $\frac{O_{n,m} q_{n,m}}{R_{k,k'}} = 0$.

1.2 效用值定义与优化问题建模

未来网络中海量机器类通信 (mMTC, massive machine type communication) 和高可靠低时延 (URLLC, ultra reliable low latency communication) 2 种典型业务对服务时延具有严格需求,因此,基于服务时延设计效用函数.

用户 u 接入第 k 个 SBS 时请求网络切片 n 中内容 m 的效用定义为

$$\ln (T_{u,n,m,k}^{\text{cac}} + T_{u,n,m,k}^{\text{com}})^{-1} \quad (8)$$

其中 $T_{u,n,m,k}^{\text{cac}}$ 和 $T_{u,n,m,k}^{\text{com}}$ 为用户完成其所请求的服务所需的总时延.

为了方便问题表述且不失一般性,假设单个切片中所有的内容大小 C_n 、计算任务卸载量 L_n 、以及计算数据回传量 O_n 均相同,分别简化表示为 C_n 、 L_n 和 O_n . 系统可以通过网络切片服务任务整形与内容块分割使得上述假设成立.

在 MEC 系统中,通过优化用户接入变量 $x_{n,u,k}$ 、缓存放置变量 $y_{u,n,m,k'}$ 与计算任务卸载变量 $z_{u,n,m,k'}$, 最大化网络切片服务效用的优化问题表示为

$$\begin{aligned} \max_{x,y,z} & \sum_{u=1}^U \sum_{n=1}^N \sum_{m=1}^{M_n} \sum_{k=1}^K x_{n,u,k} \ln (T_{u,n,m,k}^{\text{cac}} + T_{u,n,m,k}^{\text{com}})^{-1} \\ \text{s. t. } & \text{C1: } \sum_{u=1}^U \sum_{n=1}^N \sum_{m=1}^{M_n} y_{u,n,m,k'} C_n \leq C_k \\ & \text{C2: } \sum_{u=1}^U \sum_{n=1}^N \sum_{m=1}^{M_n} z_{u,n,m,k'} f_{n,m} \leq F_k \\ & \text{C3: } \sum_{k=1}^K x_{n,u,k} = 1, \forall n, \forall u \\ & \text{C4: } x_{n,u,k}, y_{u,n,m,k'}, z_{u,n,m,k'} \in \{0,1\} \end{aligned} \quad (9)$$

其中:C1,C2 和 C3 分别表示:使用的缓存容量不超过 SBS 的缓存容量,使用的计算容量不超过 SBS 的计算容量,单用户同一时刻只能接入 1 个 SBS. C4 表示 $x_{n,u,k}$ 、 $y_{u,n,m,k'}$ 和 $z_{u,n,m,k'}$ 的取值范围.

式(9)为一个非凸的非确定性多项式问题,无法直接求解. 将优化问题分解 2 个子问题后利用 SCA 交替迭代方法分别求解.

2 通信计算缓存资源联合分配算法

2.1 基于 SCA 的子问题求解

首先将优化问题式(9)分解为用户接入子问题和缓存计算资源分配子问题,并采用 SCA 方法分别实现 2 个子问题的求解.

在缓存放置变量 $y_{u,n,m,k'}$ 与计算任务卸载变量 $z_{u,n,m,k'}$ 给定的情况下,式(9)转化为用户接入子问题:

$$\begin{aligned} \min_x & \sum_{u=1}^U \sum_{n=1}^N \sum_{m=1}^{M_n} \sum_{k=1}^K x_{n,u,k} \ln (T_{u,n,m,k}^{\text{cac}} + T_{u,n,m,k}^{\text{com}}) \\ \text{s. t. } & \text{C3, } x_{n,u,k} \in \{0,1\} \end{aligned} \quad (10)$$

通过该子问题求解用户接入变量 $x_{n,u,k}$.

在用户接入变量 $x_{n,u,k}$ 给定情况下,式(9)转化为缓存计算资源分配子问题:

$$\begin{aligned} \max_{y,z} & \sum_{u=1}^U \sum_{n=1}^N \sum_{m=1}^{M_n} \sum_{k=1}^K x_{n,u,k} \ln (T_{u,n,m,k}^{\text{cac}} + T_{u,n,m,k}^{\text{com}}) \\ \text{s. t. } & \text{C1, C2, } y_{u,n,m,k'} \in \{0,1\}, z_{u,n,m,k'} \in \{0,1\} \end{aligned} \quad (11)$$

通过该子问题的求解缓存放置变量 $y_{u,n,m,k'}$ 与计算任务卸载变量 $z_{u,n,m,k'}$. 将原问题式(9)拆解为式(10)与式(11)2 个子问题交替迭代求解可以得到原问题的次优解.

将用户接入子问题式(10)的优化表达式简化,有:

$$x_{n,u,k} \ln (T_{u,n,m,k}^{\text{cac}} + T_{u,n,m,k}^{\text{com}}) = x_{n,u,k} \ln (A_{u,n,m,k'} N_k + B_{u,n,m,k'} x_{n,u,k}) - x_{n,u,k} \ln (x_{n,u,k}) \quad (12)$$

其中:

$$A_{u,n,m,k'} = \frac{y_{u,n,m,k'} C_n q_{u,n,m} + z_{u,n,m,k'} O_n q_{u,n,m}}{W_k \ln \left(1 + \frac{g_{u,k} P_k}{I_{u,k} + \sigma^2} \right)} \quad (13)$$

$$\begin{aligned} B_{u,n,m,k'} &= y_{u,n,m,k'} \frac{C_n q_{u,n,m}}{R_{k,k'}} + (1 - y_{u,n,m,k'}) \frac{C_n q_{u,n,m}}{R^{\text{B}}} + \\ & z_{u,n,m,k'} \left(\frac{O_n q_{n,m}}{R_{k,k'}} + \frac{L_n q_{u,n,m}}{f_{n,m}} \right) + (1 - z_{u,n,m,k'}) \frac{C_n q_{u,n,m}}{R^{\text{B}}} \end{aligned} \quad (14)$$

因式(12)中 $x_{n,u,k} \ln (x_{n,u,k}) \equiv 0$, 所以用户接入子问题式(10)可以转化为

$$\begin{aligned} \min_x & \sum_{u=1}^U \sum_{n=1}^N \sum_{m=1}^{M_n} \sum_{k=1}^K x_{n,u,k} \ln (A_{u,n,m,k'} N_k + B_{u,n,m,k'} x_{n,u,k}) \\ \text{s. t. } & N_k = \sum_u x_{n,u,k}, \text{C3, } x_{n,u,k} \in \{0,1\} \end{aligned} \quad (15)$$

此时引入松弛变量 $S_{u,n,m,k}$ 和 $t_{u,n,m,k'}$, 通过 SCA

方法将上式转化成式 (16), 并采用部分用户关联 (FUA, fractional user association)^[8] 的方法将 $x_{n,u,k}$ 的取值范围松弛到 $[0, 1]$, 以便于使用 SCA 的方法进行处理, 对第 1 个限制条件转化得到式 (17):

$$\begin{aligned} \min_{S, t, x} & \sum_{u=1}^U \sum_{n=1}^N \sum_{m=1}^{M_n} \sum_{k=1}^K S_{u,n,m,k} \\ \text{s. t. } & S_{u,n,m,k} \geq x_{n,u,k} t_{u,n,m,k'}, \\ & t_{u,n,m,k'} \geq \ln(A_{u,n,m,k'} N_k + B_{u,n,m,k'} x_{n,u,k}), \\ & \sum_u x_{n,u,k} = N_k, \\ & \sum_k x_{n,u,k} = 1, \\ & x_{n,u,k} \in \{0, 1\} \end{aligned} \quad (16)$$

$$\begin{aligned} S_{u,n,m,k} & \geq \\ \frac{(t_{u,n,m,k'} + x_{n,u,k})^2}{4} - \frac{t_{u,n,m,k'}^j - x_{n,u,k}^j}{2} (t_{u,n,m,k'} - x_{n,u,k}) + \\ & \frac{(t_{u,n,m,k'}^j - x_{n,u,k}^j)^2}{4} \end{aligned} \quad (17)$$

后面两项是对 $\frac{1}{4}(t_{u,n,m,k'} - x_{n,u,k})^2$ 在 $t_{u,n,m,k'}^j, x_{n,u,k}^j$ 点进行一阶泰勒展开后的结果, 其中上标 j 表示 SCA 算法进行 j 次迭代后的值. 类似地, 对第 2 个限制条件进行转化:

$$\begin{aligned} t_{u,n,m,k'} & \geq \frac{(A_{u,n,m,k'}^i N_k + B_{u,n,m,k'}^i x_{n,u,k})}{(A_{u,n,m,k'}^i N_k^j + B_{u,n,m,k'}^i x_{n,u,k}^j)} + \\ & \ln(A_{u,n,m,k'}^i N_k^j + B_{u,n,m,k'}^i x_{n,u,k}^j) - 1 \end{aligned} \quad (18)$$

这样便将式 (16) 中的前 2 个限制条件转变为凸的限制条件, 可以采用拉格朗日对偶^[9] 的方法进行求解.

对于缓存计算资源分配子问题式 (11), 同样有:

$$\begin{aligned} \min_{y, z} & \sum_{u=1}^U \sum_{n=1}^N \sum_{m=1}^{M_n} \sum_{k=1}^K x_{n,u,k} S'_{u,n,m,k'} \\ \text{s. t. } & \sum_{u=1}^U \sum_{n=1}^N \sum_{m=1}^{M_n} y_{u,n,m,k'} C_n \leq C_{k'}, \\ & \sum_{u=1}^U \sum_{n=1}^N \sum_{m=1}^{M_n} z_{u,n,m,k'} f_{n,m} \leq F_{k'}, \\ & y_{u,n,m,k'} \in [0, 1], z_{u,n,m,k'} \in [0, 1], \\ & S'_{u,n,m,k'} \geq \frac{\Delta_{u,n,m,k'}}{\Lambda_{u,n,m,k'}} + \ln(\Lambda_{u,n,m,k'}) \\ & \Delta_{u,n,m,k'} = D_{u,n,m,k'}(y_{u,n,m,k'} - y_{u,n,m,k'}^j) + \\ & D'_{u,n,m,k'}(z_{u,n,m,k'} - z_{u,n,m,k'}^j) \end{aligned} \quad (19)$$

$$\Lambda_{u,n,m,k'} = D_{u,n,m,k'} y_{u,n,m,k'}^j + D'_{u,n,m,k'} z_{u,n,m,k'}^j + E_{u,n,m,k'} \quad (21)$$

$$\begin{aligned} D_{u,n,m,k'} & = \frac{C_n q_{u,n,m}}{r_{n,u,k}} + \frac{C_n q_{u,n,m}}{R_{k,k'}} - \frac{C_n q_{u,n,m}}{R^B}, \\ E_{u,n,m,k'} & = \frac{2C_n q_{u,n,m}}{R^B} \end{aligned} \quad (22)$$

$$D'_{u,n,m,k'} = \frac{O_n q_{n,m}}{R_{k,k'}} + \frac{L_n q_{u,n,m}}{f_{n,m}} + \frac{O_n q_{u,n,m}}{r_{n,u,k}} - \frac{C_n q_{u,n,m}}{R^B} \quad (23)$$

同样, 该子问题采用拉格朗日对偶^[9] 的方法求解.

2.2 子问题交替迭代求解步骤

基于上述 2 个子问题的 SCA 求解结合 2 个子问题求解过程的迭代, 可以实现原优化问题求解. 具体通信计算缓存资源联合分配算法实现步骤如下:

初始化: 根据式 (15), 给 $y_{u,n,m,k'}, z_{u,n,m,k'}$ 随机赋初值.

步骤 1: 用户接入子问题求解: 根据式 (17)、式 (18) 的转化进行 SCA 求解, 先设置 $j=0$, 然后设置 $t_{u,n,m,k'}^j, x_{n,u,k}^j$ 和 N_k^j 的初始值并开始迭代, 每次迭代均为传统拉格朗日对偶问题的求解, 求解的值赋给本身, 执行迭代直至问题收敛, 即 $|S_{u,n,m,k}^{j+1} - S_{u,n,m,k}^j| < \epsilon$ 时, 输出 $x_{n,u,k}$.

步骤 2: 计算缓存资源分配子问题求解: 将步骤 1 得到的 $x_{n,u,k}$ 代入式 (19), 先设置 $j=0$, 然后设置 $y_{u,n,m,k'}$ 和 $z_{u,n,m,k'}$ 的初始值, 并同步骤 1 类似地进行 SCA 迭代求解, 输出 $y_{u,n,m,k'}$ 和 $z_{u,n,m,k'}$.

步骤 3: 迭代收敛判断: 将步骤 1 和步骤 2 输出的结果代入式 (9) 进行计算求出效用值 U^V , 判断与上次的计算的效用值是否满足 $|U^V - U^{V'}| > \mu$, 其中 μ 为预先设置的收敛限, 如果满足, 将步骤 2 得到的 $y_{u,n,m,k'}$ 和 $z_{u,n,m,k'}$ 作为步骤 1 的输入变量, 算法进入步骤 1; 否则算法迭代结束, 执行结果输出.

结果输出: 输出变量 $x_{n,u,k}, y_{u,n,m,k'}$ 与 $z_{u,n,m,k'}$ 的结果, 作为问题 (10) 的解.

2.3 算法收敛性和复杂度分析

定义 i 为算法交替迭代的次数, 在第 i 次迭代中, 上述 2 个子问题的第 j 次迭代在泰勒展开的局部点 $x_{n,u,k} = x_{n,u,k}^j, t_{u,n,m,k'}^j, S_{u,n,m,k}^j$ 与 U_i^V 有相同的值和梯度^[10]; 在泰勒展开的局部点 $y_{u,n,m,k'} = y_{u,n,m,k'}^j$ 和 $z_{u,n,m,k'} = z_{u,n,m,k'}^j, S_{u,n,m,k'}^j$ 与 U_i^V 有相同的值和梯度, $\{x_{n,u,k}^i, y_{u,n,m,k'}^i, z_{u,n,m,k'}^i\}, i=0, 1, \dots$ 会分别收敛至一个满足式 (11) 和式 (12) 的 Karush-Kuhn-Tucker 点. 2 个子问题都会收敛至最优解, 有 $U_i^V \geq U_{i-1}^V, \forall i \geq$

1,即发生局部收敛,且在满足前述 SCA 算法收敛条件的情况下均可达成收敛,最终在效用值变化小于收敛限的前提下,可求得一个次优的解 $U^{V[11]}$.

基于 SCA 的交替迭代算法由两部分的 SCA 迭代算法进行联合求解,直至迭代收敛至一个稳定值. 用 L 表示外层迭代的收敛次数,因为每次外层迭代都需要求解两部分 SCA 迭代算法,第一部分问题求解 1 个变量,维数为 $U \times N$,时间复杂度为 $O(U^2 N^2)$;第二部分问题求解 2 个变量,维数均为 $U \times N$,时间复杂度为 $O(UN)$. 故总时间复杂度为 $O(U^2 N^2 L)$.

3 仿真结果与分析

针对所提算法进行性能仿真验证,并与已有算法进行性能对比. 基于 Matlab 搭建了雾无线接入网系统级仿真平台,对所提算法进行了性能仿真验证与对比分析.

考虑 F-RAN 下行传输系统级仿真,1 个宏小区部署 3 个 SBS,小区半径为 200 m. 为效用值表示方便,时延以 min 为单位. 具体网络仿真参数如表 1 所示.

表 1 网络仿真参数

参数	取值
无线接入带宽 W_k/MHz	20
发射功率 P/dBm	30
缓存容量 C_k/GB	100
计算容量 $F_k/(\text{Gigacycle} \cdot \text{s}^{-1})$	1 000
噪声功率谱密度 $(\text{dBm} \cdot \text{Hz}^{-1})$	-174
切片数 N	4
回程容量 $R^B/(\text{MB} \cdot \text{s}^{-1})$	50
MEC 节点间传输能力 $R_{k,k'}/(\text{MB} \cdot \text{s}^{-1})$	20

假设系统有 4 种类型的网络切片. 不同的网络切片有不同数量的用户请求,不同切片中的用户对各自切片中内容的请求符合正态分布,计算资源值服从均匀分布,其中切片 1 和切片 2 代表低内容大小的服务,如物联网;切片 3 和切片 4 则代表高内容大小的服务,如视频服务. 具体切片仿真参数如表 2 所示.

采用典型的资源分配方式作为对比算法:随机分配算法和均匀分配算法. 随机分配算法为用户随机接入,对内容数据请求,产生 1~10 的均匀分布随机数作为有请求的用户的间隔数;对计算资源请求,产生 1~5 的均匀分布随机数作为有请求的用户的间隔数,均对 3 个 SBS 随机请求. 均匀分配算法为用户随机接入,对内容数据请求,间隔数设定为 5,对计算资源请求,间隔数设定为 3,有请求的用户按 SBS 序数对其依次请求. 显然随机分配和均匀分配

表 2 切片仿真参数

参数	切片 1	切片 2	切片 3	切片 4
用户数量	20	30	60	70
内容集容量	100	200	500	600
内容大小 C_n/MB	0.05	0.075	7	15
计算处理的数据量 $L_n/\text{Gigacycle}$	0.01	0.03	1.5	6
计算资源 $f_{nm}/(\text{Gigacycle} \cdot \text{s}^{-1})$	0.1~0.4	0.5~1	1~10	2~20
计算后的数据内容大小 O_n/MB	0.03	0.05	2	5

的算法复杂度均为 $O(UN)$.

图 2 所示为效用值随着缓存容量的变化情况. 由图可知,回程容量 R^B 为 50 MB/s,在缓存容量较小时,所提算法的效用值随着缓存容量的增大而逐渐减小,当缓存容量进一步增加到 85 GB 时,所提算法效用值则随缓存容量的增大而增大,当缓存容量逐渐增加时,资源分配策略的效果才会出现. 当 R^B 为 100 MB/s 时,采用回程会比调度的效用更高,故发生相反的情况.

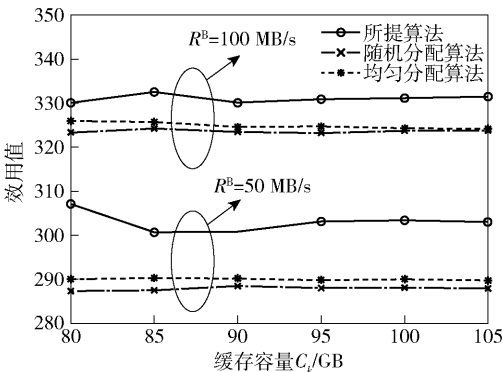


图 2 效用值随着缓存容量变化的性能对比

图 3 所示为效用值随着计算容量的变化情况,可看出与图 2 类似的趋势,即 $F_k \approx 950 \text{ Gigacycle/s}$ 时存在同样特性. 同时,由图可知,回程容量较小时,算法的增益较大,体现所提算法对回程资源的利用效率较好.

图 4 所示为效用值随着用户数的变化情况. 由图可知,随着用户数的增大,效用值减小;当用户数增大时,效用值减小的速度会变慢,且不受 R^B 的影响. 这是因为在模型设计和算法求解过程中考虑了调度和回程 2 种替代方式;用户数较小时,资源容量尚有余,效用值减小速度固定,而用户数较大时,开启调度和回程 2 种选择,效用减小速度较慢. 2 种对比算法不具此特性. 这体现了该场景下所提的资源分配方法具有较好的稳定性.

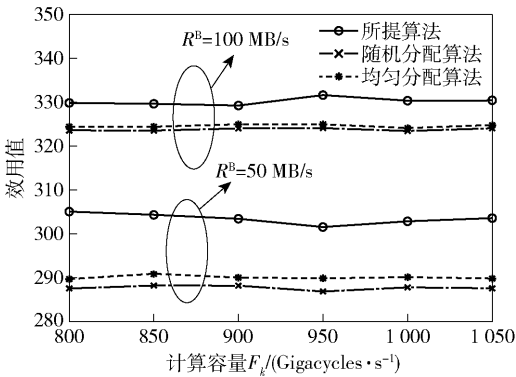


图 3 效用值随着计算容量变化的性能对比

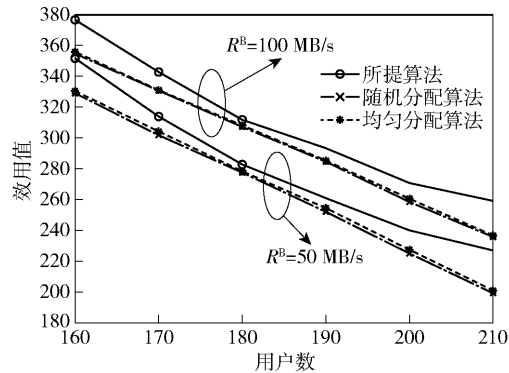


图 4 效用值随着用户数变化的性能对比

对以上数据进行对比可知,当 $C_k = 80 \text{ GB}$, $F_k = 1\,000 \text{ Gigacycles/s}$, $R^B = 50 \text{ MB/s}$ 时,所提算法的效用值相较于随机分配算法提升最多为 6.9%,相较于均匀分配算法提升了 6.2%,体现了算法性能上的优化。

4 结束语

在面向网络切片的 MEC 系统中,考虑了多 MEC 协作下的通信计算与缓存资源分配,提出了一种基于 SCA 交替迭代的联合资源分配算法。基于服务时延定义用户获得相应资源的效用值,并以用户获得资源效用值之和最大化为优化目标,通过子问题分解与 SCA 交替迭代实现了 MEC 资源分配联合优化。仿真结果表明,所提算法的效用值在随机分配算法的基础上最大提高了 6.9%,在均匀分配算法的基础上最大提高了 6.2%,表明在提高网络切片资源利用效率的基础上,实现了 MEC 系统资源联合优化的目的。

参考文献:

[1] 任品毅,许茜. 基于移动边缘计算的时延能耗最小化

安全传输[J]. 通信学报, 2020, 41(11): 52-63.

Ren Pinyi, Xu Qian. Delay and energy minimization for MEC-based secure communication [J]. Journal on Communications, 2020, 41(11): 52-63.

[2] 王侃, 赵楠, 李军怀, 等. 移动边缘计算网络中联合无线多播的服务功能链部署算法[J]. 通信学报, 2020, 41(10): 37-47.

Wang Kan, Zhao Nan, Li Junhuai, et al. Service function chain embedding algorithm with wireless multicast in mobile edge computing network [J]. Journal on Communications, 2020, 41(10): 37-47.

[3] Bi Suzhi, Huang Liang, Zhang Yingjun Angela. Joint optimization of service caching placement and computation offloading in mobile edge computing systems [J]. IEEE Transactions on Wireless Communications, 2020, 19(7): 4947-4963.

[4] Wen Wanli, Cui Ying, Quek T Q S, et al. Joint optimal software caching, computation offloading and communications resource allocation for mobile edge computing [J]. IEEE Transactions on Vehicular Technology, 2020, 69(7): 7879-7894.

[5] Ren Jianji, Wang Haichao, Hou Tingting, et al. Collaborative edge computing and caching with deep reinforcement learning decision agents [J]. IEEE Access, 2020, 8: 120604-120612.

[6] Wang Yue, Gu Yu, Tao Xiaofeng. Edge network slicing with statistical QoS provisioning [J]. IEEE Wireless Communications Letters, 2019, 8(5): 1464-1467.

[7] Rezvani S, Parsaeefard S, Mokari N, et al. Cooperative multi-bitrate video caching and transcoding in multicarrier NOMA-assisted heterogeneous virtualized MEC networks [J]. IEEE Access, 2019, 7: 93511-93536.

[8] Ye Qiaoyang, Rong Beiyu, Chen Yudong, et al. User association for load balancing in heterogeneous cellular networks [J]. IEEE Transactions on Wireless Communications, 2013, 12(6): 2706-2716.

[9] Boyd S, Boyd S P, Vandenberghe L. Convex optimization [M]. London: Cambridge University Press, 2004: 207-223.

[10] Zeng Yong, Zhang Rui. Energy-efficient UAV communication with trajectory optimization [J]. IEEE Transactions on Wireless Communications, 2017, 16(6): 3747-3760.

[11] Yang Yang, Pesavento M, Chatzinotas S, et al. Successive convex approximation algorithms for sparse signal estimation with nonconvex regularizations [J]. IEEE Journal of Selected Topics in Signal Processing, 2018, 12(6): 1286-1302.