

基于线性规划松弛的移动边缘计算卸载模型

雷雪梅, 刘丽, 王倩

引用本文

雷雪梅, 刘丽, 王倩. [基于线性规划松弛的移动边缘计算卸载模型](#)[J]. 计算机科学, 2023, 50(6A): 211200229-5.

LEI Xuemei, LIU Li, WANG Qian. [MEC Offloading Model Based on Linear Programming Relaxation](#)[J]. Computer Science, 2023, 50(6A): 211200229-5.

相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

[天然气智能调度计算机平台系统的设计与实现](#)

Design and Implementation of Natural Gas Intelligent Scheduling Computer Platform System
计算机科学, 2023, 50(6A): 220700258-7. <https://doi.org/10.11896/jsjcx.220700258>

[中继选择和队列稳定动态能量优化策略](#)

Dynamic Energy Optimization Strategy Based on Relay Selection and Queue Stability
计算机科学, 2023, 50(6A): 220100082-8. <https://doi.org/10.11896/jsjcx.220100082>

[面向能耗优化和负载均衡的边缘服务器放置研究](#)

Edge Server Placement for Energy Consumption and Load Balancing
计算机科学, 2023, 50(6A): 220300088-5. <https://doi.org/10.11896/jsjcx.220300088>

[6G重叠区域中基于博弈论的任务卸载策略](#)

Task Offloading Strategy Based on Game Theory in 6G Overlapping Area
计算机科学, 2023, 50(5): 302-312. <https://doi.org/10.11896/jsjcx.220500120>

[基于拍卖的边缘云期限感知任务卸载策略](#)

Auction-based Edge Cloud Deadline-aware Task Offloading Strategy
计算机科学, 2023, 50(4): 241-248. <https://doi.org/10.11896/jsjcx.211200194>

基于线性规划松弛的移动边缘计算卸载模型

雷雪梅¹ 刘丽² 王倩²

1 北京科技大学信息化建设与管理办公室 北京 100083

2 北京科技大学自动化学院 北京 100083

(xmlei@ustb.edu.cn)

摘要 移动边缘计算中本地设备可以将计算任务卸载到靠近网络的边缘节点上执行,只将计算结果回传至用户端,从而减小传输时延,降低移动设备的功耗,减轻客户端的负载压力,还可以减少核心网络的计算负载。针对复杂多类边缘节点的移动边缘计算环境,建立了基于线性规划松弛的三级计算任务卸载决策模型,称为 CART-CRITIC-LR(CCLR)。首先通过分类回归决策树算法(CART)筛选出本地执行的计算任务;然后采用多属性决策的 CRITIC 算法确定 3 个性能指标的权值分配;最后,基于线性规划松弛(LR)对计算卸载问题建模,使计算任务卸载决策的总时延、总能耗和总成本最优。实验比较了其他计算卸载策略的能耗、成本、延迟时间等指标,结果表明 CCLR 卸载决策算法在保证多目标全局最优的同时,实现了总时延最短,说明了所提算法的有效性与适用性。

关键词: 移动边缘计算;任务卸载;多属性决策;分类回归决策树;线性规划

中图分类号 TP393

MEC Offloading Model Based on Linear Programming Relaxation

LEI Xuemei¹, LIU Li² and WANG Qian²

1 Office of Information Construction and Management, University of Science and Technology Beijing, Beijing 100083, China

2 School of Automation and Engineering, University of Science and Technology Beijing, Beijing 100083, China

Abstract In the mobile edge computing(MEC), the local device can offload tasks to the edge node near the network for computation processing, thereby reducing the delay, power consumption and overload of the client, also the computing loading core network. For the complex MEC environment of multi-type edge nodes, a three-stage computing offloading decision is modeled based on linear programming relaxation, that is CART-CRITIC-LR(CCLR) algorithm. First, the classification and regression decision tree algorithm(CART) is used to screen out the locally executed calculation tasks. Secondly, the multi-attribute decision-making algorithm(CRITIC) is used to determine the weight of the three performance indicators respectively. Then the calculation offloading problem is modeled as a linear programming relaxation(LR) to optimize the equilibrium solutions among the total delay, total energy consumption and total cost. Each offloading strategy is analyzed by comprehensively comparing the energy consumption, cost, delay. experimental results show that the CCLR algorithm achieves the shortest total delay while ensuring the multi-objective global optimization, which illustrates the effectiveness and applicability of the algorithm.

Keywords Mobile edge computing, Task offloading, Multi-attribute decision, Classification and regression tree, Linear programming

1 引言

随着 5G 时代的到来,移动智能设备的应用量大幅增长,移动用户对延迟敏感型和计算密集型应用程序的需求逐渐加大,如虚拟现实、交互式游戏等^[1]。结合物联网技术的飞速发展,“智慧+”产业的应用越发广泛^[2]。然而,这种复杂的应用程序会比以往在移动设备上执行的应用消耗更多资源,用户设备依靠其有限的计算资源和电池容量无法满足计算任务的需求,移动边缘计算(Mobile Edge Computing, MEC)作为有效的解决方案被提出。

MEC 将计算任务迁移到边缘执行,只将处理结果回传至用户端,计算卸载不仅可以减小传输时延、降低移动设备的能耗、减轻客户端的负载压力,还可以减少核心网络的计算负载^[3]。计算卸载对网络的服务质量有着很大影响,如何在 MEC 网络框架下作出计算卸载决策是一个挑战。国内外研究学者对计算卸载问题进行了一些研究,主要解决了面向多用户优化时延或能耗的计算卸载问题。文献[4]研究了保证服务性能的任务卸载调度问题,把移动边缘计算的系统优化问题视为移动用户的能量消耗最小化问题,并以资源和时延作为约束。文献[5]提出了多用户的能量感知计算卸载

基金项目:国家自然科学基金面上项目(12071025);佛山市科技创新专项资金项目(BK20AE004)

This work was supported by the National Natural Science Foundation of China(12071025)and Scientific and Technological Innovation Foundation of Foshan Municipal People's Government(BK20AE004).

通信作者:刘丽(liuli@ustb.edu.cn)

机制,在信道干扰阈值和时间期限约束下,将卸载决策问题归结为一个 0-1 非线性整数规划问题,将能耗最小作为优化目标,对移动设备进行分类和优先级的确定,基于逆向拍卖原理设计了定价模型,实现了计算卸载算法。Zhang 等考虑了移动用户的异构性和资源的有限性,针对多个服务提供商、多个移动设备的场景,提出了多环密封顺序组合拍卖算法(MSS-CA)^[6]。Cui 等提出了一种面向移动边缘计算的多用户任务卸载方法^[7],将计算任务表示为有向无环图,基于改进 NS-GA-II 建立了计算卸载多目标优化模型。Liang 等在移动边缘计算环境下针对多用户多任务单卸载站点的场景提出了分布式基于深度学习的卸载算法^[8]。El Haber 等将聚合节点与 MEC 边缘服务器区分开,提出了一种基于逐次凸逼近法的低复杂度算法来处理 MEC 环境中的资源分配和任务卸载问题^[9]。文献^[10]提出了一种异构网络中的分布式联合计算卸载和资源分配优化算法(JCORAO),通过博弈论求解最佳的计算卸载策略策略。Fan 等提出了一种车联网边缘计算环境下的基于深度强化学习的协同计算卸载方案来优化任务时延^[11]。

以上研究大多考虑的是单一类别的边缘节点,或优化单一目标。然而在边云异构多任务、多边缘、结点复杂的 MEC 的情境下的计算卸载问题研究更符合实际应用,为了优化多个目标,本文提出了一种三阶段计算卸载决策模型 CCLR (CART-CRITIC-LR)。该模型首先通过分类与回归决策树算法(Classification And Regression Tree, CART)筛选出在移动设备本地执行的计算任务,降低算法整体的复杂度;然后采用多属性决策的独立性权重(Criteria Importance Though Intercrieria Correlation, CRITIC)算法确定 3 个优化目标的权值分配;最后将 MEC 环境下的计算卸载问题建模为线性规划问题,以最小化总延迟、总能耗和总成本的加权和为目标,利用基于松弛理论的线性规划求解计算卸载的位置。

2 计算卸载决策模型

2.1 MEC 网络架构

边缘节点特指那些将 MEC 服务器部署在接近移动用户位置的网络节点。不同的边缘节点可能有不同的计算资源以及 CPU 频率,致使它们为用户提供处理性能和使用体验也会有所差别。边缘节点大多部署在无线基站上,无线基站的异构性决定了不同边缘节点之间的差别。3GPP 组织根据基站的单载波发射功率和覆盖能力将无线基站分为 4 类:宏基站、微基站、皮基站和飞基站。本文选择宏基站和微基站作为 MEC 中二类边缘节点,宏基站作为边缘节点规模较大,具有更多的资源,在云服务提供区域上也比微基站大,其服务范围可基本覆盖多个微基站的服务区域。在宏基站和微基站之间,存在回程将微基站的传输中继到宏基站。

本文根据边缘节点的类别提出了一个可扩展的网络架构,包括远端云、多类的边缘节点和移动设备,如图 1 所示。其中移动设备通过无线网络与边缘节点连接,如 wifi, 3G, 4G 等。不同类的边缘节点之间通过光纤连接;而边缘节点通过高速有线网络与远端云连接。

可扩展的多类边缘云框架包括任务集、移动设备、二类边缘节点以及远端云。所有计算任务的执行位置有 4 种选择:移动设备本地、微基站、宏基站以及远端云。计算任务首先

需要考虑是否在移动设备本地执行,本地设备的计算性能和电池容量普遍有限,但由于其物理距离为零,传输延迟也不存在。若移动设备的性能足以提供计算任务所需的资源,则优先本地执行,否则卸载到边缘节点进行执行。在同样满足覆盖区域内的边缘节点中,计算任务需要选择带来最优性能指标的网络节点。如果任务卸载到微基站但其资源无法满足任务所需,则将其卸载到资源更丰富的宏基站。

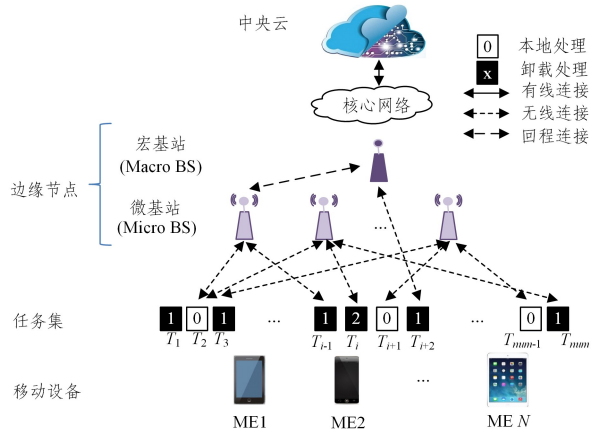


图 1 多类边缘节点的 MEC 网络架构

Fig. 1 Architecture of MEC with different kinds of edge nodes

2.2 计算卸载建模

(1) 任务模型

移动用户根据需要会在移动设备上产生若干个计算任务,其属性受应用程序和移动用户的需要影响。任务的数据大小为 α_i 、MEC 服务器返回的数据大小为 β_i ,以及任务所需计算资源为 γ_i 。用整数变量 x 表示执行计算任务的卸载位置,变量等于 1 表示计算任务被卸载到 MEC 服务器 k 处。整数变量 x 如式(1)所示,若 x 无解则卸载至远端云执行。

$$x_{ik} = \begin{cases} 0, & \text{不卸载至 MEC 服务器 } k \\ 1, & \text{卸载至 MEC 服务器 } k \end{cases}, \forall i \in num, k \in \{0, 1, 2, \dots, K+1\} \quad (1)$$

(2) 计算模型

假设 f_0 表示本地移动设备的 CPU 频率,若计算任务在移动设备本地执行,则任务 i 的计算延迟为 $T_{i0}^{\text{comp}} = \frac{\gamma_i}{f_0}$;假设 f_k 表示边缘节点的 CPU 频率,若计算任务不在移动设备本地执行,则任务 i 的计算延迟为 $T_{ik}^{\text{comp}} = \frac{\gamma_i}{f_k}$;假设移动设备的执行功率为 p_{local} ,边缘节点的执行功率为 p_{edge} 。由此可得移动设备和 MEC 边缘节点执行任务时消耗的总能耗为:

$$E^{\text{comp}} = \sum_{k=1}^{K+1} \sum_{i=1}^{num} (p_l + p_e) \cdot (T_{ik}^{\text{comp}} + T_{i0}^{\text{comp}}) \quad (2)$$

(3) 通讯模型

假设有 num 个计算任务,它们来自 N 个移动设备,且部署了 K 个微基站以及 1 个宏基站。本文假设边缘节点和移动设备在特定区域中满足泊松分布,并将边缘节点和移动设备之间的距离设定为 d_{nk} 。由此可知,任务卸载到边缘节点后,边缘节点和移动设备之间的信道增益为 $h_{nk} = 103.8 + 20.9 \cdot \log_{10} d_{nk}$ 。边缘节点和移动设备之间的上行和下行的传输速率为:

$$\begin{aligned} C_k^{\text{UL}} &= B_k^{\text{UL}} \log_2 \left(1 + \frac{P^{\text{UL}} h_{nk}^{\text{UL}}}{\omega_0} \right) \\ C_k^{\text{DL}} &= B_k^{\text{DL}} \log_2 \left(1 + \frac{P^{\text{DL}} h_{nk}^{\text{DL}}}{\omega_0} \right) \end{aligned} \quad (3)$$

将宏基站与微基站之间的回程延迟记作 T^B , 将边缘节点和计算任务之间的上行和下行传输延迟分别表示为 T_{ik}^{UL} 和 T_{ik}^{DL} , 如式(4)所示:

$$\begin{aligned} T_i^B &= 0.001 \cdot \alpha_i \\ T_{ik}^{UL} &= \frac{\alpha_i}{C_k^{UL}} + T_i^B \\ T_{ik}^{DL} &= \frac{\beta_i}{C_k^{DL}} + T_i^B \end{aligned} \quad (4)$$

因此, 通讯所需总能耗如式(5)所示:

$$E^{\text{comm}} = P^r \sum_{k=1}^{K+1} \sum_{i=1}^{mm} x_{ik} T_{ik}^{UL} + P^r \sum_{k=1}^{K+1} \sum_{i=1}^{mm} x_{ik} T_{ik}^{DL} \quad (5)$$

(4) 卸载决策模型

本文将总能耗、总时间延迟和总成本作为卸载决策的优化指标。总能耗包括通讯能耗和计算能耗, 如式(6)所示:

$$E(X) = E^{\text{comm}} + E^{\text{comp}} \quad (6)$$

总时间延迟由传输延迟和计算延迟组成, 如式(7)所示:

$$T(X) = \max_{k \in K} (T_{ik}^{UL} + T_{ik}^{DL} + T_{ik}^{\text{comp}}) \quad (7)$$

计算任务的数据大小决定了在执行过程中的成本, 若微基站、宏基站的单位数据的执行成本分别表示为 m_{MBS} 和 m_{MBS} , 则总执行成本如式(8)所示:

$$M(X) = m_{\text{MBS}} \sum_{k=1}^K \sum_{i=1}^{mm} x_{ik} \alpha_i + m_{\text{MBS}} \sum_{i=1}^{mm} x_{i(K+1)} \alpha_i \quad (8)$$

计算卸载问题即如何得到卸载决策 x , 从而使总时间延迟、总能耗和总成本三者满足约束的条件下达到最小。本文研究的优化问题模型也是一个带约束的多目标最小化问题, 计算卸载决策的约束条件是每个任务只能卸载到一个位置, 如式(9)所示:

$$\begin{aligned} \min \{ & \lambda_1 T(X) + \lambda_2 E(X) + \lambda_3 M(X) \} \\ \text{s. t. } & \sum_{k=1}^{K+1} x_{ik} = 1, \forall i \in \text{num}, x_{ik} \in \{0, 1\} \end{aligned} \quad (9)$$

3 基于松弛的三阶段计算卸载算法

移动网络产生的数据量呈指数级增长, 这给接入网络带来了巨大压力。目前已有的基于进化算法的计算卸载模型未考虑不同计算任务对各个性能指标的需求^[12]。例如, 运算复杂型应用对计算资源的需求会比对时间指标的要求更高。针对总延迟、总能耗和总成本这3个性能指标, 本文通过优化权重分配处理卸载决策问题, 提出三级计算任务卸载决策 CCLR 算法, 即任务筛选、权重分配和线性规划。

3.1 任务筛选

对计算任务进行筛选可以区分出移动设备本地资源足以满足的任务, 这些任务无须卸载到更高的网络节点增加网络负载, 同样也简化了后续卸载位置的分析, 从而降低了算法的总复杂度。

本文采用决策树分类的思想来处理计算任务, 实现对任务进行筛选。决策树算法适用于中小样本集, 提取的特征满足不相关性, 同时具有计算复杂度低、可解释性强的优势。决策树分类器是一种有监督的机器学习方法, 有多种的生成算法, 如 ID3, C4.5 和分类回归决策树算法(CART)等。

CART 算法采取了基尼系数作为特征选取的准则, 概率分布的基尼系数如式(10)所示。而基尼系数 $Gini(S, A)$ 代表经过特征 $A = a$ 分割后集合 S 的不确定性, 如式(11)所示:

$$Gini(S) = \sum_{i=1}^n p_i (1 - p_i) = 1 - \sum_{i=1}^n p_i^2 \quad (10)$$

$$Gini(S, A) = \frac{|S_1|}{|S|} Gini(S_1) + \frac{|S_2|}{|S|} Gini(S_2) \quad (11)$$

基尼系数衡量数据集的不纯度, 选出不纯度最低的特征, 即基尼系数最小的特征可以达到更好的分类效果^[13]。与 ID3 算法和 C4.5 算法不同, CART 算法在计算过程中不必进行信息熵的对数运算, 因此运行更高效, 同时也可以处理连续型数据。

CART 构成了二叉决策树, 并且可以处理标量特征和连续型特征。CART 作为一种决策树, 由根节点、分支节点、叶节点和有向边组成, 如图 2 所示。图中用正方形代表根节点, 用圆形代表叶节点, 用圆角矩形表示分支节点, 并且注明了该节点的分类或回归规则, 即是否满足某个属性特征的要求。在分支节点, 只有满足与否两种取值可能, 其本质是对特征空间进行二元划分。叶节点则代表了分类或回归的结果, 在图 2 中, 以类别 1 和类别 2 为例, 每个叶节点实则代表从根节点到该叶子节点的路径所满足的某套规则。例如, 满足“分类或回归规则 2”且满足“分类或回归规则 4”的叶节点属于类别 2。

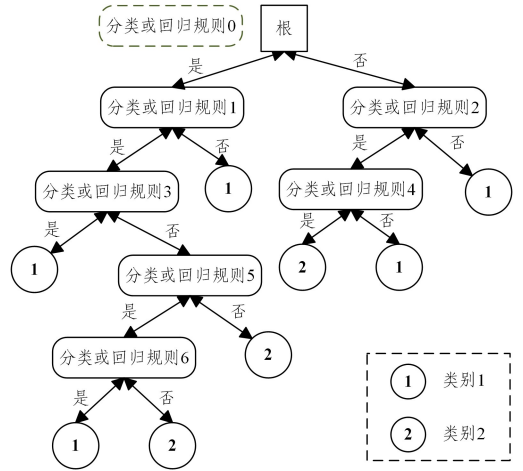


图 2 CART 示意图

Fig. 2 Schematic diagram of CART

选择 CART 算法生成决策树对计算任务进行分类。本文的问题模型中存在 5 个样本特征可供分析, 分别是任务数据量、任务回传数据量、任务所需 CPU 计算资源、移动设备与边缘节点的距离和信道增益。

3.2 权重分配

卸载决策的优化目标包括总时间延迟、总能耗和总成本。各优化目标权值会影响最终卸载决策的结果, 本文将权重分配的问题转换为多属性决策问题。

CRITIC 方法作为一种客观赋值法, 其基本思路是以对比强度和指标之间的冲突性为基础^[14]。对比强度是指同一指标在各个方案下取值的差距大小, 以标准差的形式来表现, 该数值越大, 各方案的取值差距越大, 对比强度越强。指标之间的冲突性是以指标之间的相关性为基础, 若指标之间呈较强正相关, 说明指标冲突性较低。CRITIC 的计算步骤如算法 1 所示。

算法 1 CRITIC 算法

输入: 数据矩阵 A

输出: 权重 W

1. 数据清洗: 奇异点检测(3 σ 原则)、无量纲化处理(极大型指标 $s_{ij} =$

$$\frac{f_{ij} - \min f_{ij}}{\max f_{ij} - \min f_{ij}}; \text{极小型的指标 } s_{ij} = \frac{\max f_{ij} - f_{ij}}{\max f_{ij} - \min f_{ij}}$$

2. 计算相关系数: 相关系数是反映变量之间的相关关系密切程度的统计指标。以变量 i 和 j 为例, 按积差方法计算, 以两变量与各自平均值的差为基础, 通过相乘来反映两变量的相关程度。

$$r_{ij} = \frac{\sum (i - \bar{i})(j - \bar{j})}{\sqrt{\sum (i - \bar{i})^2 \sum (j - \bar{j})^2}}$$

3. 计算第 j 个指标的冲突性量化指标: $\sum_{i=1}^n (1 - r_{ij})$

4. 计算标准差: $\delta_j = \sqrt{\frac{\sum_{i=1}^m (r_{ij} - \bar{r}_j)^2}{m}}$, 其中 \bar{r}_j 为第 j 个属性的平均值。

5. 计算信息量 C : 第 j 个评价指标所包含的信息量 $C_j = \delta_j \sum_{i=1}^n (1 - r_{ij})$

6. 计算权重: $W_j = \frac{C_j}{\sum_{j=1}^m C_j}$

3.3 基于线性规划松弛的算法

通过不同赋权方法得到的权重分配可以确定总目标函数, 优化模型如式(12)所示:

$$\begin{aligned} \min \lambda_t T(X) + \lambda_e E(X) + \lambda_m M(X) \\ \text{s. t. } \sum_{k=1}^{K+1} x_{ik} = 1, \forall i \in \text{num}, x_{ik} \in \{0, 1\} \end{aligned} \quad (12)$$

由于 $T(X)$ 在计算过程中需要取最大化, 故引入新变量 t , 如式(13)所示:

$$t \geq T_{ik}^{\text{UL}} + T_{ik}^{\text{DL}} + T_{ik}^{\text{comp}} \quad (13)$$

故对优化问题进行转化, 如式(14)所示:

$$\begin{aligned} \min \lambda_e E(X) + \lambda_t t + \lambda_m M(X) \\ \text{s. t. } T_{ik}^{\text{UL}} + T_{ik}^{\text{DL}} + T_{ik}^{\text{comp}} \leq t \end{aligned} \quad (14)$$

$$\sum_{k=1}^{K+1} x_{ik} = 1, \forall i \in \text{num}, \forall k \in (K+1), x_{ik} \in \{0, 1\}$$

通过线性松弛后, 新自变量为 $y = [x^T, t]^T$ 。为了方便进行线性规划, 借助系数矩阵, 将模型整理为标准的线性规划形式, 如式(15)所示^[15]:

$$\begin{aligned} \min b_0^T y \\ \text{s. t. } A_1 y \leq 0_{(K+1) \times 1} \\ A_2 y = 1_{m \times 1} \\ y_j \in [0, 1], \forall j = 1, \dots, \text{num} \cdot (K+1) \end{aligned} \quad (15)$$

其中系数矩阵分别可表示为:

$$\begin{aligned} b_0 &= [\lambda_t b_1^T, \lambda_m b_2^T, \lambda_e]^T \\ b_1 &= P^r [T_{K+1}^{\text{UL}T}, T_{K+1}^{\text{DL}T}, \dots, T_{K+1}^{\text{UL}T}]^T + P^e [T_{K+1}^{\text{DL}T}, T_{K+1}^{\text{DE}T}, \dots, \\ & T_{K+1}^{\text{DL}T}]^T + (p_t + p_e) \cdot [T_{K+1}^{\text{DE}T}, T_{K+1}^{\text{DE}T}, \dots, T_{K+1}^{\text{DE}T}]^T \\ T_k^{\text{UL}} &= [T_{1k}^{\text{UL}}, \dots, T_{mnk}^{\text{UL}}]^T, T_k^{\text{DL}} = [T_{1k}^{\text{DL}}, \dots, T_{mnk}^{\text{DL}}]^T \\ T_k^{\text{DE}} &= [T_{1k}^{\text{DE}}, \dots, T_{mnk}^{\text{DE}}]^T, T_{ik}^{\text{DE}} = T_{ik}^{\text{comp}} + T_{ik}^{\text{B}} \\ b_2 &= [M_1^T, M_2^T, \dots, M_{K+1}^T]^T, M_k = [M_{1k}, \dots, M_{mnk}]^T \\ M_{ik} &= \alpha_i \cdot m_{ik}, m = [m_{\text{MBS}}, \text{ones}(1, K), m_{\text{MBS}}] \end{aligned}$$

$$A_1 = \begin{bmatrix} D_1^T, 1_{0_1 \times \text{num}}, \dots, 0_{1 \times \text{num}}, -1 \\ 0_{1 \times \text{num}}, D_2^T, \dots, 0_{1 \times \text{num}}, -1 \\ \vdots \\ 0_{1 \times \text{num}}, 0_{1 \times \text{num}}, \dots, D_{K+1}^T, -1 \end{bmatrix},$$

$$\begin{aligned} D_k &= [D_{1k}, \dots, D_{mnk}]^T, D_{ik} = T_{ik}^{\text{UL}} + T_{ik}^{\text{DL}} + T_{ik}^{\text{comp}} \\ A_2 &= [I_{\text{num}, 1}, I_{\text{num}, 2}, \dots, I_{\text{num}, K+1}, 0_{\text{num} \times 1}] \end{aligned} \quad (16)$$

其中, $I_{\text{num}, k}$ 表示 num 维的单位矩阵。

由此可知, 本文提出的三阶段计算卸载算法包括 CART 算法和 CRITIC 法和线性规划松弛 (Linear programming Relaxation, LR)。该计算卸载算法 CART-CRITIC-LR (CCLR) 算法的流程如算法 2 所示。

算法 2 CCLR 计算卸载算法

输入: $\text{num}, K, \alpha_i, \beta_i, \gamma_i, f_{ik}, c_k^{\text{UL}}, c_k^{\text{DL}}, \forall i \in \text{num}, \forall k \in K+1$

输出: X^{LR}

1. 通过 CART 算法进行任务筛选;
2. 通过 CRITIC 法确定各优化目标的权值, 如算法 1 所示;

3. 确定线性规划, 并求解得 \hat{X} ;

4. If \hat{X} 是二进制矩阵, then $X^{\text{LR}} = \hat{X}$;

Else

For $i=1$; num do

$k^* = \arg \max_{k \in K+1} \hat{x}_{ik}$;

$x_{ik}^{\text{R}} = 0, \forall k \in K+1 \setminus \{k^*\}, x_{ik}^{\text{R}} = 1$;

End for

End if

4 CCLR 算法性能评价

将所提出的 CCLR 算法与熵值法和 AHP 算法的权值分配结果、整数规划算法、基于 LR 的计算卸载算法以及多目标进化算法进行卸载性能比较。实验在 MATLAB 2018b 上进行仿真验证, 并使用了 Classification Learner 工具箱。实验环境为 Inter(R) Core(TM) CPU i5-4570S (2.90 GHz 和 8G RAM)。

4.1 实验准备

本文假设了 2 类边缘节点, 包括 3 个微基站和 1 个宏基站, 其分布位置、计算能力和拥有资源的程度不同。设定移动设备的本地执行频率为 0.6×10^9 cycles/s, 微基站的 CPU 频率为 1.0×10^{10} cycles/s, 宏基站的 CPU 频率为 8.0×10^{10} cycles/s。在计算成本时, 设定微基站的单位执行成本 $m_{\text{MBS}} = 0.1$ \$, 宏基站的单位执行成本 $m_{\text{MBS}} = 0.12$ \$。在能耗方面, 设定移动设备的接收功率和发射功率均为 0.2 W。移动设备在工作时的功率 p_{local} 大小为 150 mW, 空闲时的功率大小为 355 mW; 而 MEC 边缘节点在工作时的功率 p_{edge} 大小为 800 mW, 在空闲时功率大小为 100 mW。

计算任务的数据大小由 α 表示, 服从 $[10, 30]$ MB 的均匀分布; 从 MEC 服务器回传的数据大小占原数据大小的 20%, 即 $\beta = 0.2 \times \alpha$; 且任务所需计算资源 $\gamma = q \times \alpha$, 其中 q 由其引用类型决定, 本文取 $q = 330$ 。在 CART 算法中设定参数如下: 分支数目为 20; K 折交叉验证数为 5; 训练样本和测试样本均为 80。

4.2 计算卸载性能分析

将 CCLR 算法与多目标进化算法 MOEA/D, NSGAI, 以及 MOEA/D-CMA 进行对比。从图 3 可以看出, 随着任务数目从 30 增加到 120, 总延迟、总能耗、总成本及其加权和都基本实现了数值上的逐步上升。CCLR 算法在具备良好的扩展性的同时, 实现了性能指标加权和的最优和总延迟性能上的最优。当任务集较大时, 例如任务数等于 120, CCLR 算法在总成本上性能不如其他算法, 但由于算法更关注的是整体的

综合效果,故略高的成本并不影响整体加权和性能。并且在处理较大的任务集时,CCLR算法仍可给出较优解,使加权和保持在稳定的较低数值,具备良好的扩展性。

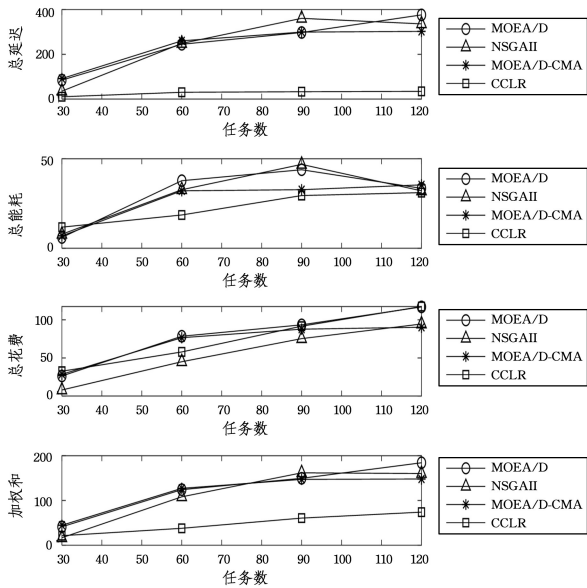


图3 CCLR算法与多目标算法的性能对比

Fig. 3 Performance comparison of CCLR and other multi-objective algorithm

结束语 本文研究了多类边缘节点的 MEC 环境的计算卸载问题。根据边缘节点的处理性能、覆盖范围等特征将边缘节点细化为宏基站与微基站,并考虑二者之间的传输所带来的回程延迟。建立了该场景下的计算卸载决策模型,主要包括任务模型、通讯模型和计算模型,同时提出了一种三级任务卸载决策 CCLR 算法。该算法首先通过分类回归决策树算法(CART)筛选出在移动设备本地执行的计算任务,降低算法整体的复杂度;然后采用多属性决策的 CRITIC 算法确定 3 个优化目标的权值分配,以满足不同应用对性能指标的需求;最后将计算卸载问题建模为线性规划问题,以最小化总延迟、总能耗和总成本的加权和为目标,利用基于松弛理论的线性规划求解计算卸载的位置,并与 0-1 整数规划和多种多目标优化算法进行对比。实验结果表明,CCLR 算法在计算任务的数目增加时算法性能较优,其总时延、总能耗和总成本的加权和最优,并且在总延迟性能上优势更为突出。

参考文献

- [1] CHEN X, CAI Y, LI L, et al. Energy-Efficient Resource Allocation for Latency-Sensitive Mobile Edge Computing [J]. IEEE Transactions on Vehicular Technology, 2019, 69(2): 2246-2261.
- [2] LIU X, YANG Q, LUO J, et al. An energy-aware offloading framework for edge-augmented mobile RFID systems [J]. IEEE Internet of Things, 2018, 6(3): 3994-4004.
- [3] WANG S, ZHANG X, ZHANG Y. A survey on mobile edge networks: Convergence of computing, caching and communications [J]. IEEE Access, 2017, 5: 6757-6779.
- [4] TAO X, OTA K, DONG M, et al. Performance Guaranteed Computation Offloading for Mobile-Edge Cloud Computing [J].

IEEE Wireless Communications Letters, 2017, 6(6): 774-777.

- [5] LAN L, XIAOYONG Z, KAIYANG L, et al. An Energy-Aware Task Offloading Mechanism in Multiuser Mobile-Edge Cloud Computing [J]. Mobile Information Systems, 2017, 5(7): 13455-13464.
- [6] ZHANG H, GUO F, JI H, et al. Combinational Auction Based Service Provider Selection in Mobile Edge Computing Networks [J]. IEEE Access, 2017, 5: 13455-13464.
- [7] CUI Y Y, ZHANG D G, ZHANG T, et al. A Multi-User Fine-Grained Task Offloading Scheduling Approach of Mobile Edge Computing [J]. Acta Electronica Sinica, 2021, 49(11): 2202-2207.
- [8] HUANG L, FENG X, FENG A, et al. Distributed Deep Learning-based Offloading for Mobile Edge Computing Networks [J]. Mobile Networks and Applications, 2022, 27(6): 1123-1130.
- [9] EL HABER E, NGUYEN T M, ASSI C. Joint Optimization of Computational Cost and Devices Energy for Task Offloading in Multi-Tier Edge-Clouds [J]. IEEE Transactions on Communications, 2019, 67(5): 3407-3421.
- [10] ZHANG J, XIA W, YAN F, et al. Joint Computation Offloading and Resource Allocation Optimization in Heterogeneous Networks With Mobile Edge Computing [J]. IEEE Access, 2018: 19324-19337.
- [11] FAN Y F, YUAN S, CAI Y, et al. Deep Reinforcement Learning-based Collaborative Computation Offloading Scheme in Vehicular Edge Computing [J]. Computer Science, 2021, 48(5): 270-276.
- [12] SUN G, AYEPAH-MENSAH D, LU L, et al. Delay-aware content distribution via cell clustering and content placement for multiple tenants [J]. Journal of Network & Computer Applications, 2019, 137(2019): 112-126.
- [13] SMAYRA T, CHARARA Z, SLEILATY G, et al. Classification and Regression Tree(CART) model of sonographic signs in predicting thyroid nodules malignancy [J]. European Journal of Radiology Open, 2019, 6: 343-349.
- [14] DIAKOULAKI D, MAVROTAS G, PAPAYANNAKIS L. Determining objective weights in multiple criteria problems: The critic method [J]. Computers & Operations Research, 1995, 22(7): 763-770.
- [15] DINH T Q, TANG J, LA Q D, et al. Offloading in Mobile Edge Computing: Task Allocation and Computational Frequency Scaling [J]. IEEE Transactions on Communications, 2017, 65(8): 3571-3584.



LEI Xuemei, born in 1972, Ph.D, senior engineer. Her main research interests include mobile computing, network optimization and data analysis.



LIU li, born in 1968, Ph.D, professor. Her main research interests include mobile computing and multi-objective optimization.