

文章编号:1007-5321(2021)01-0072-07

DOI:10.13190/j.jbupt.2020-115

基于任务间依赖关系的小小区协作卸载策略

康曼聪, 李 曦, 纪 红, 张鹤立

(北京邮电大学 信息与通信工程学院, 北京 100876)

摘要: 在密集部署的小小区网络中,考虑到小小区基站(SBS)的计算资源有限,提出了基于任务间串并依赖关系的协作卸载策略,以降低计算卸载任务的整体完成时延。首先,考虑将可以同时执行的并行任务卸载至不同的 SBS,利用计算资源的分布式特点来降低整体时延,同时最大化单个 SBS 上的串行任务数量,以减小所需 SBS 的数目;然后,根据网络的负载均衡情况对 2 种场景进行讨论,联合考虑任务间的依赖关系、不同 SBS 的可用计算资源量和 SBS 与用户间的信道质量,分别引入最长路径理论和图着色算法以确定最佳任务卸载方案。仿真结果表明,与已有策略相比,所提策略可降低计算卸载任务的整体完成时延。

关键词: 小小区网络; 移动边缘计算; 协作卸载; 时延优化

中图分类号: TN929.5

文献标志码: A

Collaborative Computation Offloading Exploring Task Dependencies in Small Cell Networks

KANG Man-cong, LI Xi, JI Hong, ZHANG He-li

(School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract: In dense small cell networks, a task-dependency-based collaborative offloading scheme is proposed to cope with the limited computational resources in small cell base station (SBS), which can further reduce the total execution delay. Firstly, the parallel tasks are offloaded to different SBSs to reduce computing delay, while sequential tasks are offloaded to the same SBSs to reduce the number of required SBS. Then, based on whether different SBSs have the same load pressure or not, the optimal offloading strategies are proposed in two different scenarios by introducing the longest path theory and the graph coloring theory, respectively. The strategies jointly consider the user energy constraint, dependencies among tasks, the amount of available computational resources and channel conditions of different SBSs. Simulation shows that the proposed strategies can reduce total execution delay compared with existing strategies in both scenarios.

Key words: small cell networks; mobile edge computing; collaborative computation offloading; latency optimization

在未来小小区网络中,小小区基站(SBS, small cell base station)部署密集且计算资源有限。将用户

应用分解为多个任务并卸载至不同的 SBS 进行计算可有效提升小小区计算卸载服务质量^[1]。然而,

收稿日期: 2020-08-14

基金项目: 国家自然科学基金项目(61771070)

作者简介: 康曼聪(1995—),女,博士生。

通信作者: 李 曦(1983—),女,教授,博士生导师, E-mail: lixi@bupt.edu.cn。

如何根据该网络中独特的计算资源分布特点来设计高效的卸载策略还有待深入研究。

首先,卸载策略应充分考虑任务间的串并依赖关系。当今,越来越多的应用开始采用并行架构,例如虚拟现实和人脸识别^[2]。将其中互为并行关系的任务卸载至不同的 SBS 有两方面的优点:一方面,位于相同 SBS 上的任务将存在串行依赖关系,不需要共享或排队使用计算资源,由此可削弱单个 SBS 资源较少带来的影响;另一方面,不同节点通过同时执行并行任务可有效降低整体时延,使得网络中 SBS 的数量优势得以展现。同时,为减小所需的 SBS 数量和信令开销,需要最大化单个 SBS 上的串行任务数量。此外,卸载过程需考虑不同任务的比特大小和所需的计算量,以实现 SBS 间负载均衡^[3],降低卸载传输时延和能耗^[4]。

现有文献已在移动边缘计算(MEC, mobile edge computing)领域及云计算领域开展了与协作卸载有关的研究。在 MEC 网络中,为减小计算时延和用户本地能耗,学者将计算任务分解成任意大小的比特包或相互独立的子任务,卸载至不同边缘节点进行计算,例如宏基站^[4]、SBS^[5]、无人机以及云一边服务器等。学者进一步将任务间的依赖关系纳入考虑范围。例如,He 等^[6]将深度神经网络分解成多个串行依赖任务,卸载到不同边缘计算节点以最小化整体时延。而 Ding 等^[3]则考虑任务间为一般性依赖关系(包含串并关系),将多个任务协作卸载至不同的 MEC 服务器以平衡时延和能耗开销。此外,在云计算中,Sahoo 等^[7]以最小化系统开销为目标,考虑任务间的串并关系,为每个任务选择最佳的物理服务器进行计算。

然而,MEC 网络中的大部分策略,假设卸载节点可以无限制地同时处理多个任务,或者没有利用任务间的并行关系来降低整体时延^[1-3]。而云计算中的大部分策略则没有将任务的发放过程纳入考虑范围^[7]。据此,以最小化整体任务完成时延为目标,提出了充分利用任务间串并依赖关系的协作卸载策略。为了降低通信时延,引入了非正交多址接入技术(NOMA, none-orthogonal multiple access),同时同频地将任务发往各个 SBS。仿真结果表明,基于 NOMA 的策略与基于频分多址接入技术(FDMA, frequency division multiple access)的策略相比,在时延方面有明显改善。考虑到不同 SBS 的负载情况对卸

载过程的影响,基于 2 种负载场景进行分析,并分别引入最长路径理论和图着色算法以找到最优卸载方案。仿真结果表明,在 2 种场景下,与已有策略相比,所提策略可进一步降低整体时延。

1 系统建模

1.1 网络模型

未来网络中将部署大量的 SBS 为用户提供服务^[1,5]。用户可将应用分解成多个任务后发往不同的 SBS 进行计算。考虑到卸载传输时延一般远大于信道相关时间,根据瞬时信道增益的期望值(简称为“信道增益”)来设计卸载策略。设用户和第 k 个 SBS 之间的信道增益为 g_k ,将 SBS 按照信道增益大小进行排序: $g_1 > g_2 > \dots > g_k > \dots > g_{K^{\max}} > g_{th}$,有 K^{\max} 个 SBS 的信道质量满足信道增益阈值 g_{th} 的要求。根据 SBS 间是否具有相同的负载压力,为 2 种场景设计最佳卸载策略。对一个典型用户的卸载过程进行分析,未来工作中将对多用户场景进行讨论。

一个应用中任务之间的依赖关系可以用调用图 $G = (V, E)$ 表示。其中, V 为节点集合, $j \in V$ 是其中的一个任务节点, E 为有向边的集合,有向边 $e_{i,j} \in E$ 代表任务间的依赖关系。当任务 i 的输出是任务 j 的输入时, $e_{i,j} \neq 0$,反之, $e_{i,j} = 0$ 。在 1.2 节和 2.1 节中, $e_{i,j}$ 的数值大小将分别等于端点任务的计算时延和比特量大小。图 1 给出了一个调用图的例子。值得注意的是,原始调用图往往包含很多对卸载过程可以忽略不计影响的任务节点^[2],例如“系统库对象”,一般将他们从调用图中删除以简化卸载过程。在所得调用图中,节点数目将比较小(例如人脸识别应用仅有 5 个节点^[2]),设计集中式的卸载策略依然可以满足实时性要求。所提到的调用图均指已删除多余节点的调用图。此外,为了让策略更具有普适性,所研究的调用图基于一般性依赖关系模型,并且只有一个起始节点和末尾节点,分别对应于应用中负责接收用户输入的模块和将结果展示给用户的模块,必须在用户本地执行。

在图 1 中,与任务 7 呈并行关系的任务分别为任务 5,6,8,应该避免将它们与任务 7 卸载至同一个 SBS。同时,与任务 7 呈串行关系的任务包括任务 1,2,3,4,9,应该最大化地将它们与任务 7 发往同一个 SBS。为了便于对时延进行建模,根据任务

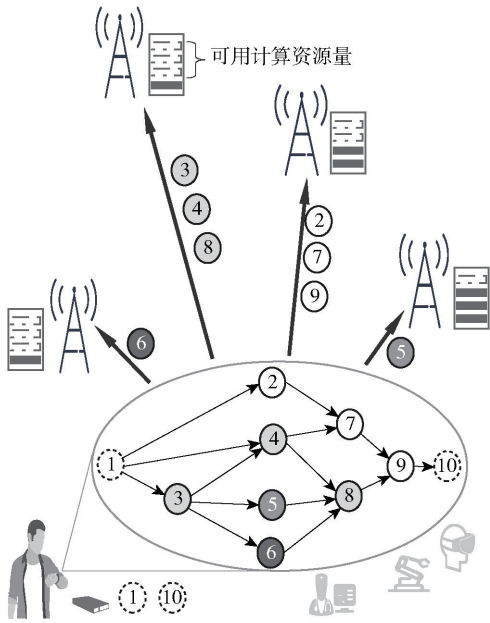


图1 考虑任务间依赖关系的协作卸载策略

依赖关系对任务节点进行拓扑排序。另外,设调用图中的节点数目为 N 。任务 $j(j=2, \dots, N-1)$ 的卸载选择方案用 $a_{k,j}$ 表示, $a_{k,j} \in A$, A 为包含了全部任务的卸载选择方案的集合。 $a_{k,j}=1$ 表示将任务 j 卸载至第 k 个 SBS, 否则 $a_{k,j}=0$ 。由于并行任务不可被卸载到相同的 SBS, 若任务 i 和任务 j 相互并行, 则 $a_{k,i}a_{k,j}=0$ 。用户本地只计算始末节点, $a_{0,1}=a_{0,N}=1$ 。将在算法部分确定所需的最小 SBS 数目, 记为 K , 并使用前 K 个 SBS 来服务用户。分别使用 μ_j 和 φ_j 来表示任务 j 的比特大小和计算量, 他们之间的关系为 $\varphi_j = \omega_j \mu_j$, 其中 ω_j 为任务 j 中每比特所需的计算量大小。

1.2 计算模型

分别使用 f_0 和 f_k 来表示用户本地和第 k 个 SBS 的可用计算资源量大小。由于并行任务将被发往不同的 SBS, 每个任务都可使用所在 SBS 的全部可用资源进行计算, 因此任务 $j \in \{1, 3, \dots, N\}$ 的计算时延为: $D_j^c = \sum_{k=0}^K a_{k,j} \frac{\varphi_j}{f_k}$ 。每个任务必须要等到它所依赖的全部任务计算完毕后才开始计算。令调用图中有向边 $e_{i,j}$ 的数值大小等于任务 j 的计算时延, 则任务 j 计算完毕的时间等于由起始节点到节点 j 的最长路径值 $l(j)$, 其计算如下:

$$l(j) = \begin{cases} \frac{\varphi_1}{f_0}, & j=1 \\ \max_{i \in \{i | e_{i,j} \neq 0\}} \{l(i) + D_j^c\}, & 1 < j \leq N \end{cases} \quad (1)$$

由式(1)可知, 到某节点的最长路径值只取决于到其所有父节点的最长路径值。可使用动态规划算法, 依次计算到每个节点的最长路径值。总体计算时延 D^c 等于到末尾节点的最长路径值, 时间复杂度为 $O(n)$, n 为节点数目。此外, 根据计算能耗与 CPU 主频之间的关系^[3], 若用户本地有效电容系数为 κ_u , 则用户本地的计算能耗为: $E_0^c = (\varphi_1 + \varphi_N) \kappa_u f_0^2$ 。

1.3 通信模型

为了进一步降低通信时延, 采用 NOMA 将各任务同时同频地卸载至相应的 SBS。对于很多应用而言, 任务间传输的计算结果比特量远小于任务卸载过程中的比特大小^[8], 例如“图像处理应用”, 因此忽略任务间的通信时延。接下来给出卸载传输时延关于传输能耗的表达式。

Wu 等^[4]将一个计算任务分割成任意大小的比特包, 基于 NOMA 技术发往不同的 SBS 进行计算, 并给出了总体最小传输功率和传输时延之间的关系表达式, 见式(2)。该表达式的推导过程可以概括为: 根据香农公式和给定的最小传输速率, 由近到远依次计算用户向各基站的最小发射功率, 然后求和。忽略不同小区之间的干扰。

$$P^t(s, D^t) = W \left(\sum_{k=1}^K \left(\frac{n_0}{g_k} - \frac{n_0}{g_{k-1}} \right) 2^{\frac{1}{D^t} \sum_{i=k}^K s_i} - \frac{n_0}{g_K} \right) \quad (2)$$

其中: D^t 为给定的传输时延; W 为用户带宽; $s \triangleq (s_k)_{k \in \{1, 2, \dots, K\}}$ 为卸载比特向量, $s_k = \sum_{j=2}^{N-1} a_{k,j} \mu_j$ 为卸载至第 k 个 SBS 的总任务比特大小; 参数 $g_0 \rightarrow +\infty$; n_0 为噪声功率谱密度。传输能耗为 $E^t(s, D^t) = P^t(s, D^t) D^t$ 。当给定 E^t 和 s 时, 可使用二分法解得 D^t 。

1.4 问题建模

首先, 推导在用户能耗和功率限制下最小整体时延的表达式, 然后, 给出问题建模表达式。根据 1.3 节, 卸载传输时延是关于卸载方案和传输能耗的函数。事实上, 在用户能耗和功率限制下, 可以得到仅随卸载方案变化的最小传输时延表达式。接下来展开推导过程。

推导在用户能耗限制下的最小传输时延表达式。需要证明传输时延随能耗的增加而单调递减。根据 1.3 节, 对 NOMA 传输能耗 E^t 求关于传输时延 D^t 的一阶偏导:

$$\frac{\partial E^l(s, D^l)}{\partial D^l} = Wn_0 \left\{ \sum_{k=1}^K \left(\frac{1}{g_k} - \frac{1}{g_{k-1}} \right) \left(1 - \frac{\ln 2}{W D^l} \sum_{i=k}^K s_i \right) 2^{\frac{1}{D^l} \sum_{i=k}^K s_i} - \frac{1}{g_K} \right\} \quad (3)$$

继续对 E^l 求关于 D^l 的二阶偏导:

$$\frac{\partial^2 E^l(s, D^l)}{\partial (D^l)^2} = \sum_{k=1}^K \frac{n_0 (\ln 2)^2}{W (D^l)^3} 2^{\frac{1}{D^l} \sum_{i=k}^K s_i} \left(\frac{1}{g_k} - \frac{1}{g_{k-1}} \right) \left(\sum_{i=k}^K s_i \right)^2 \quad (4)$$

上式结果显然大于 0, 表明一阶偏导随着 D^l 的增加单调递增. 又因 $\lim_{D^l \rightarrow +\infty} \frac{\partial E^l(s, D^l)}{\partial D^l} = \left(-\frac{Wn_0}{g_0} \right)$,

可得对于任何 s , 都有 $\frac{\partial E^l(s, D^l)}{\partial D^l} < 0$. 由此可知, 在

卸载方案一定的情况下 (s 的值一定), E^l 随着 D^l 的增大而单调递减. 根据反函数的性质, D^l 也随着 E^l 的增大而单调递减, 当 E^l 取最大值 ($E^{\max} - E_0^c$) 时可以得到 D^l 的最小值, 其中 E^{\max} 为用户本地能量大小. 因此, 使用二分法对 E^l 的表达式进行求解, 即可得到满足用户能耗限制的最小传输时延 $D_1^{l, \min}$.

除了满足用户能耗限制之外, 还需满足发射功率限制 P^{\max} . 根据式 (2), 可以得到当传输时延为 $D_1^{l, \min}$ 时所需发射功率大小. 若其大于 P^{\max} , 则需要根据功率限制重新计算最小传输时延. 此时, 根据式 (2) 和反函数的性质, 可以得到传输时延随功率的增加而单调递减. 因此, 可使用二分法求解式 (2), 得到在功率 P^{\max} 下的最小传输时延 $D_2^{l, \min}$. 同时考虑用户能耗限制和功率限制的最小传输时延表达式为

$$D^l(A) = \begin{cases} D_1^{l, \min}, & P^l(s, D_1^{l, \min}) \leq P^{\max} \\ D_2^{l, \min}, & P^l(s, D_2^{l, \min}) > P^{\max} \end{cases} \quad (5)$$

最终, 以最小化时延为目标的最优化问题可以被建模为最小化问题:

$$A^{\text{opt}} = \arg \min_A (D^c + D^l) \quad (6)$$

$$\text{s. t. C1: } \sum_{k=1}^K a_{k,j} = 1, \forall j = 2, \dots, N-1$$

$$\text{C2: } a_{0,1} = a_{0,N} = 1$$

$$\text{C3: } a_{k,i} a_{k,j} = 0, \text{ 若任务 } i, j \text{ 相互并行}$$

$$\forall k = 1, \dots, K$$

其中用户能耗和功率的限制条件已在 D^l 的推导过程中得到满足. 限制条件 C1 到 C3 为任务选择计算节点的约束条件, 已在 1.1 节中详细说明.

2 考虑任务间关系的协作卸载策略

根据 SBS 间的不同负载情况, 分别为 2 种典型场景提出基于任务间串并关系的最优协作卸载策略, 以在用户能耗受限的情况下最小化整体时延.

2.1 场景一: SBS 间负载压力相同

当 SBS 间负载压力相同时, 不同 SBS 的可用计算资源量相同. 此时, 所有卸载任务将具有相同的计算速度, 总计算时延 D^c 将是一个定值. 而对于传输时延, 由 Wu 等^[4] 向信道质量好的 SBS 卸载的总比特量越大, 卸载传输时延将会越小.

因此, 在此场景下, 需要最大化卸载至排在前面的 SBS 的任务比特量. 为此, 令调用图中的 $e_{i,j}$ 等于任务 j 的比特大小, 得到最优卸载方案的过程为: 使用动态规划算法从调用图中依次找到当前最长路径, 并分别把每条最长路径上的节点向序号从小到大排列的 SBS 上卸载, 具体过程如图 2 所示. 在该方案中, 并行任务不可能位于同一条路径上, 因此将被卸载至不同的 SBS, 同时实现最大化在同一个 SBS 上的串行任务数目. 可用反证法证明算法的最优性: 假设存在另一种卸载方案可以获得更小的整体时延, 则该方案等价于在上述最优卸载方案的基础上, 对一些任务的计算位置进行调整, 这将会导致向信道质量差的 SBS 所卸载的比特量增加, 致使整体时延变大, 与假设矛盾.

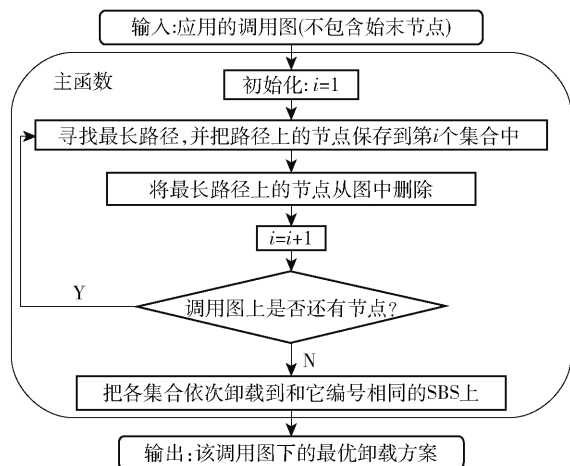


图 2 场景一中基于最长路径理论的协作卸载策略

2.2 场景二:SBS 间负载压力不同

当用户周围各 SBS 的负载压力不同且在动态变化时,它们的可用计算资源量各不相同,总计算时延和传输时延大小都将随卸载方案变化.在此场景下,引入图着色算法来寻找最优卸载方案,如图3所示.

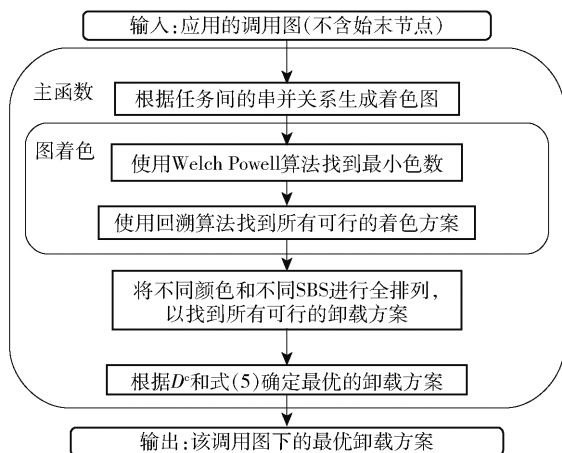


图3 场景二中基于图着色算法的协作卸载策略

首先,用户根据本地应用的调用图生成“着色图”.在着色图中,根据调用图中任务的串并关系,令并行任务相连,串行任务不相连.然后,将相连的节点涂上不同的颜色.第一步,使用经典的 Welch Power 算法以较低的复杂度确定着色所需最小色数的近似值(即 K 值).第二步,引入回溯算法使用 K 种颜色对着色图进行着色,以找到所有可能的着色方案.接下来,用户设备向周围的 SBS 发起请求以获取 SBS 的状态信息,包括信道状态和可用计算资源量大小.然后,用户设备将 K 种颜色和前 K 个 SBS 进行全排列,以找到所有可行的卸载方案.最后,根据 D^c 和式(5)确定使整体时延最小的最优卸载方案,并向 SBS 发出资源占用请求,将有着相应颜色的任务卸载到对应的 SBS 上.值得注意的是,在用户收集完 SBS 信息之后的算法部分的执行时间极短,适合在 SBS 负载动态变化的环境下使用.此外,由于最优方案是基于遍历算法得到的,证明了所提策略的最优性.需要说明的是,由于调用图中的节点数目较小,着色时所需色数 K 将比较小.同时,在小小区网络中存在比较多的 SBS 满足与用户连接要求^[1].因此,可以认为最终卸载过程将使用 K 个 SBS 进行卸载.

2.3 算法复杂度分析

在场景一中使用动态规划算法,从图中寻找最

长路径以获得最优解.由1.2节知,其时间复杂度为 $O(N)$.在场景二中,使用 Welch Power 算法和回溯算法来获得着色方案的可行解集,他们的时间复杂度分别为 $O(N^2)$ 和 $O(NK^N)$.然后,将颜色和 SBS 进行全排列以获得全部可行卸载方案,时间复杂度为 $O(K!)$.因此,场景二的复杂度为 $O(NK^N)$,它随着任务数的增加呈指数增长.由1.1节知,由于原始调用图中删除多余节点后,调用图中的节点数将比较少,所提策略可用于大多数应用的情况.

3 仿真结果与讨论

对所提策略的性能进行评估.在小小区网络中,假设用户周围有4个SBS的信道质量满足连接要求^[1].每个任务的比特大小取0.1~1MB.依次生成节点数为1到20的应用调用图(不含多余节点),以体现绝大多数应用中的任务情况^[2].在每个节点数目下,为避免局限于现有文献中的几种应用结构,随机生成具有一般性依赖关系的50幅不同的调用图,并记录各策略下他们的平均整体时延大小.经测试,所生成的调用图在使用着色算法时,最小色数均小于等于4,可以使用所提策略完成卸载工作(详细过程参见2.2节).其他仿真参数见表1.

表1 仿真参数表

参数	参数值
W/MHz	8
$n_0/(\text{dBm} \cdot \text{Hz}^{-1})$	-174
κ_u	10^{-28}
$\omega_f/(\text{CPU cycles} \cdot \text{byte}^{-1})$	4 000
P^{\max}/W	1
E^{\max}/J	1
g_1, g_2, g_3, g_4	$10^{-4}, 10 \cdot 3^{-4}, 11^{-4}, 13^{-4}$
$f_0, f_1, f_2, f_3, f_4/(\text{GHz})$	1, 15, 10, 9, 8

首先,为证明将用户任务发往多个SBS进行协作卸载的必要性,第一个对比算法“单SBS策略”来自Guo等所提策略^[8],考虑将所有任务发往同一个SBS进行排队计算.该策略遍历用户周围所有SBS,并记录整体时延的最小值.接着,为证明将并行任务卸载到不同的SBS可以有效利用小小区网络中分布式的计算资源来降低整体时延,第二个对比算法“双SBS策略”来自Ding等所提策略^[3],考虑多个任务仅可以被卸载到任意2个SBS上(基于NO-

MA)进行计算. 当多个并行任务位于同一个 SBS 上时,将其分配到最少的不同虚拟机中以便同时执行. 虚拟机之间平分计算资源. 同样,在周围的 SBS 上遍历所有可行的卸载方案,并记录整体时延的最小值. 仿真将验证,所提策略通过把并行任务发往不同的 SBS,能够获得较低的整体任务执行时延.

各策略下的平均整体时延如图 4 和图 5 所示. 由图可知,各策略下的平均整体时延均随任务数目的增大而增大. 当任务数目较少时,2 个及以下的 SBS 便可满足所提策略的卸载需求,使得双 SBS 策略将拥有与所提策略相同的最优卸载方案和整体时延. 并且由于可以同时执行并行任务,该整体时延将小于排队执行并行任务的单 SBS 策略. 当任务数目较多时,2 个 SBS 不能满足所提策略的卸载需求. 此时,在双 SBS 策略下,并行任务将被分配到不同虚拟机中以保证同时执行. 这将导致单个任务所占用的计算资源减少,使得该策略的整体时延将大于所提策略. 随着任务数目的进一步增长,虚拟机数目将持续增加,整体执行时延不断增大. 当任务数目到达一定程度时,任务速度减小对整体时延产生的影响甚至超过了任务并行计算所产生的影响,如图 5 所示,双 SBS 策略的时延可能会超过排队计算所有串并行任务的单 SBS 策略.

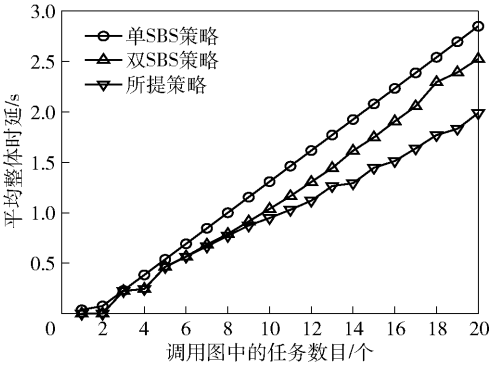


图 4 场景一下 3 种策略的平均整体时延对比

图 6 所示为在不同用户能耗限制下^[9]的整体时延大小比较. 使用 E 代表 E^{\max} . 根据 1.3 节, NOMA 的卸载时延随着能耗的增大而单调递减,因此当用户本地能量越小时,整体时延越大.

图 7 所示为基于 NOMA 与基于 FDMA 的策略对比. 在 FDMA 下,将用户带宽平分给各目标 SBS, 剩余的策略部分与所提策略相同. 由图 7 可知,随着任务数目的增加,基于 NOMA 的策略将会获得更

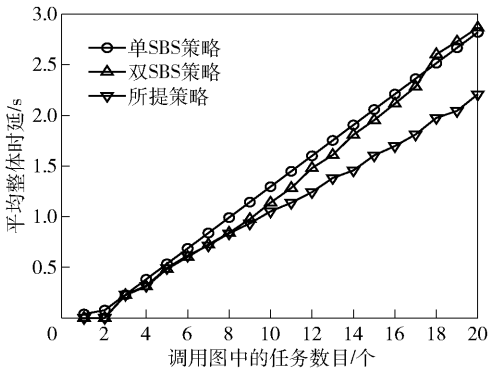


图 5 场景二下 3 种策略的平均整体时延对比

短的时延.

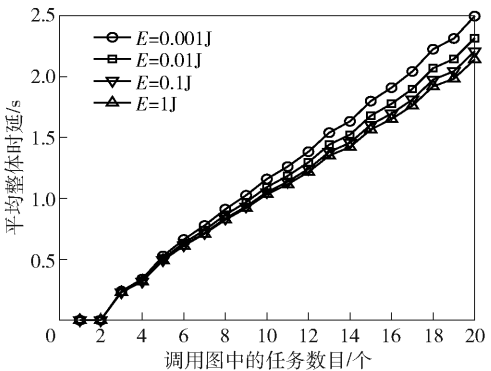


图 6 不同能耗限制下的平均整体时延对比

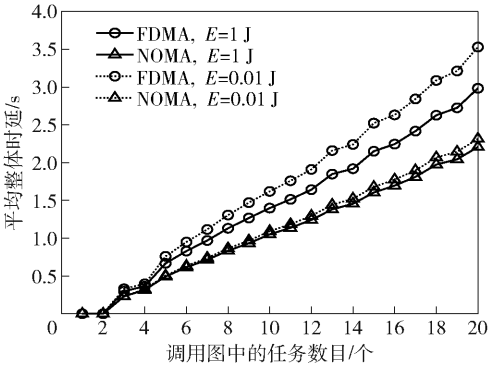


图 7 基于不同传输机制的平均整体时延对比

4 结束语

针对小小区网络中的计算资源分布特点,在 2 种典型场景下,综合考虑 SBS 的负载压力和信道质量,将并行任务卸载至不同的 SBS,串行任务卸载至相同的 SBS,提出使整体时延最小化的协作卸载策略. 仿真结果表明,所提策略可降低小小区网络中计算卸载任务的整体完成时延. 未来工作将继续在多用户场景中对基于任务间依赖关系的协作卸载策

略展开研究,并引入博弈论对用户间的竞争关系进行讨论.

参考文献:

- [1] Liu Yiming, Yu F R, Li Xi, et al. Joint access and resource management for delay-sensitive transcoding in ultra-dense networks with mobile edge computing[C]//ICC 2018. Kansas City: IEEE Press, 2018: 1-6.
- [2] Niu Jianwei, Song Wenfang, Atiquzzaman M. Bandwidth-adaptive partitioning for distributed execution optimization of mobile applications[J]. Journal of Network and Computer Applications, 2014, 37: 334-347.
- [3] Ding Yan, Liu Chubo, Zhou Xu, et al. A code-oriented partitioning computation offloading strategy for multiple users and multiple mobile edge computing servers[J]. IEEE Transactions on Industrial Informatics, 2020, 16(7): 4800-4810.
- [4] Wu Yuan, Ni Kejie, Zhang Cheng, et al. NOMA-assisted multi-access mobile edge computing: a joint optimization of computation offloading and time allocation[J]. IEEE Transactions on Vehicular Technology, 2018, 67(12): 12244-12258.
- [5] Seng Shuming, Luo Changqing, Li Xi, et al. User matching on blockchain for computation offloading in ultra-dense wireless networks[J/OL]. IEEE Transactions on Network Science and Engineering, 2020(2020-06-09), [2020-11-16]. <https://ieeexplore.ieee.org/document/9112693>. DOI: 10.1109/TNSE.2020.3001081.
- [6] He Wenchen, Guo Shaoyong, Guo Song, et al. Joint DNN partition deployment and resource allocation for delay-sensitive deep learning inference in IoT[J]. IEEE Internet of Things Journal, 2020, 7(10): 9241-9254.
- [7] Sahoo P K, Dehury C K, Veeravalli B. LVRM: on the design of efficient link based virtual resource management algorithm for cloud platforms[J]. IEEE Transactions on Parallel and Distributed Systems, 2018, 29(4): 887-900.
- [8] Guo Songtao, Liu Jiadi, Yang Yuanyuan, et al. Energy-efficient dynamic computation offloading and cooperative task scheduling in mobile cloud computing[J]. IEEE Transactions on Mobile Computing, 2019, 18(2): 319-333.
- [9] Wang Feng, Xu Jie, Ding Zhiguo. Multi-antenna NOMA for computation offloading in multiuser mobile edge computing systems[J]. IEEE Transactions on Communications, 2019, 67(3): 2450-2463.