An Overhead-Optimizing Task Scheduling Strategy for Ad-hoc Based Mobile Edge Computing

Li Tianze, Wu Muqing, Zhao Min, Liao Wenxing

**一、本文贡献：**

**基于ad hoc的移动边缘计算系统**，提出了一种**开销优化的多设备任务调度策略**。该任务调度策略考虑了**opportunity consumption, time delay, energy consumption, and monetary cost**，旨在**最小化每个移动设备的开销**。

1.提出了基于ad-hoc的移动边缘计算系统模型，分析了移动设备的开销。

2.将任务调度问题描述为**一个分布式多设备任务调度博弈**。通过构造一个势函数，证明了**任务调度博弈是一个势博弈**，它具有有限的改进性质，且总是具有纳什均衡。

3.设计了一种开销优化的多设备任务调度算法。

4.进行仿真以评估所提出策略的效果。

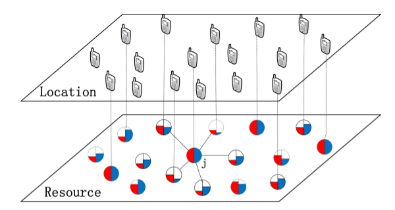
结果表明，所提出的任务调度策略能有效地降低移动设备的开销，并成功地完成任务。

**二、系统模型：**

移动边缘计算系统分为**基于基础设施的移动边缘云**和**基于ad hoc的移动边缘虚拟云**。

基于基础设施的云：由**远程服务器**授权，可以为移动设备提供高效的资源。

基于ad-hoc的虚拟云：**由一组移动设备组成**，这些移动设备协同工作以完成任务

**本文基于ad-hoc的移动边缘计算系统。(ad-hoc架构如下)**

基于ad-hoc的移动边缘计算系统实例。红色（蓝色）部分表示可用带宽（计算）资源。

移动设备j是一个资源丰富的设备，它的邻居设备可以使用它的资源。

在ad-hoc系统中，任务调度问题的**挑战:**

1)设备的移动设备的接触持续时间变化，任务调度策略必须考虑接触持续时间。

2)任务调度涉及**额外的数据传输**，太多的设备同时通过无线接入来加载任务，会对彼此造成严重的**干扰**，从而降低数据传输速率并导致低能量效率。

3)计算资源不能自由使用，当使用另一台设备的资源时，任务发布者应**付费**作为奖励。

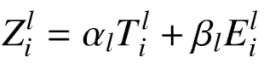
**1.任务执行模型**

**1) Local Execution:**

**任务执行时间：**

F^l\_i:移动设备i的计算能力 ;

D\_i:任务的工作负载 ; D^w\_i:等待在移动设备i中执行的任务的工作负载

**能耗：** v\_i:每个执行单元的能耗

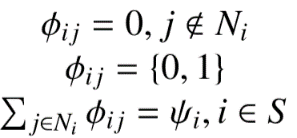
**本地计算的开销:**

αl,βl是权重因子，0≤αl，βl≤1，αl+βl=1。**不同的设备选不同的权重因子**

**2) Offloaded Execution:**

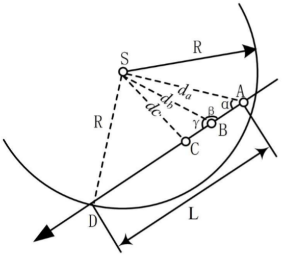
**Ψi**表示移动设备中有任务需要执行为1，否则为0。**φ\_ij**表示移动设备i的任务加载到其相邻设备j为1，否则为0。**φ\_ii**指示移动设备i是否本地执行其任务。

移动设备i的邻居表示为**Ni={j，k，…}**，Ni中的每个移动设备可直接与i通信。

约束： 确保可以将任务加载到与i无线连接的移动设备上。

表示任务可以在本地执行或加载到附近的设备上。

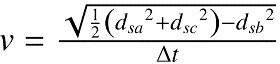
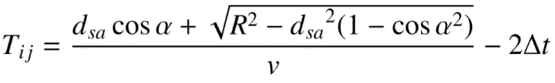
表示所有移动设备的任务应仅安排一次。

**2.移动设备的接触持续时间**

移动设备i和j之间的相对移动示例图->

假设移动设备i在点A和移动设备j在初始时刻s点处

移动设备i和j之间的接触持续时间为：

****

R是移动设备i和j之间的无线链路的最大距离，它们之间的相对运动速度为v

**3.通信和计算模型**

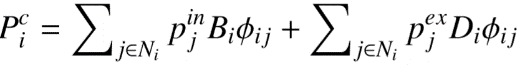
移动设备j的邻居设备Nj的任务调度决策，φ\_Nj=（φ\_lj,φ\_kj,…,φ\_nj）

**移动设备i在加载任务时会产生一些消耗:**

**1)time consumption：**T^c\_n = 任务的输入数据发送时间 t^s\_ij + 在附近设备上执行任务的时间t^ex\_ij +接收输出数据的时间 t^r\_ij

**2)energy consumption:** E^c\_n = 数据发送能量 E^s\_ij + 数据接收能量 E^r\_ij

**3)monetary consumption:** 设移动设备j一字节的输入数据支付p^in\_j，per execution unit 支付p^{ex}\_{v\_ci}，（忽略了输出数据的monetary cost）

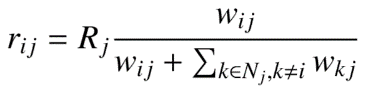
****

忽略了移动设备接收输出数据的能量和时间消耗，即tr ij=0，Er ij=0。

因此，移动设备的总时间消耗T^c\_i和总能耗表示为：



F\_ij:由移动设备j分配给移动设备i的计算资源

P^s\_i:设备i的传输功率

r\_ij:无线信道上移动设备i和j之间的期望吞吐量

Rj表示移动设备的最大传输速率j

Wij是表示无线竞争信道中的移动设备i的权重以访问移动设备j的整数。

**移动设备i的总开销:**

αc、βc、γc和为1，并且根据移动设备的需求，每个因子可以设置不同的值

**三、Task Offloading Analysis**

移动设备i在N\_j中的**所有任务调度决策表示为φ\_{Nj\i}=（φ\_kj，φ\_lj，…，φ\_Nj）**

如果φ\_ii = 1, Zi = Z^l\_i，如果φ\_ii = 0, Zi = Z^c\_i.

上述问题可以表述为一个开销优化的多设备任务调度博弈

φ\*\_ij 为

决定φ\*\_ij应满足以下四个约束：

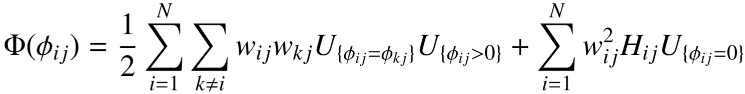
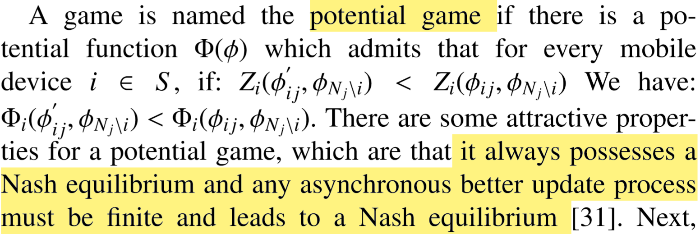
**1） opportunity consumption 约束**。只有那些机会消耗小于阈值的移动设备才能提供资源，这是为了在移动设备中没有足够的资源时避免资源消耗。

**2） Overloading约束。**分配给j的所有任务应小于j的剩余计算资源

**3）延迟容限**。接触持续时间应该大于总时间消耗： ，执行时间应小于时间延迟限制Tmax。

**4） 利益约束**。

如果在均衡中没有移动设备可通过改变自己的策略来进一步降低其开销，那么strategy profile是纳什均衡。开销优化的多设备任务调度博弈是一个**势博弈**，该博弈存在**纳什均衡**。



**四、算法设计**

在这种任务调度机制中，每个移动设备根据从其邻居设备接收到的这些参数进行分析。

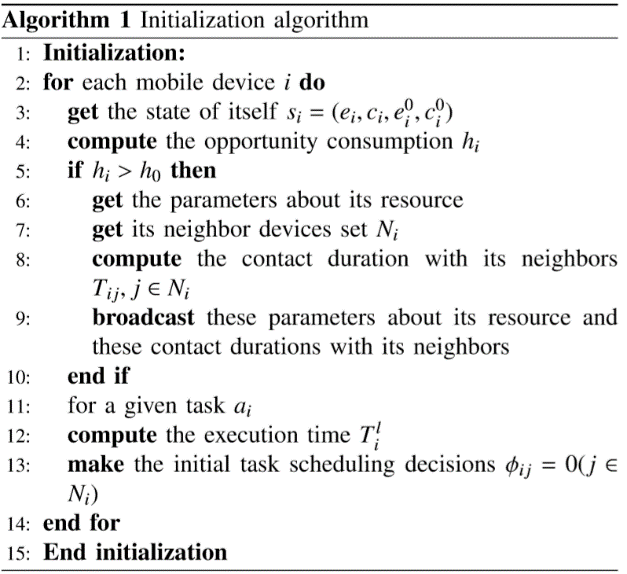
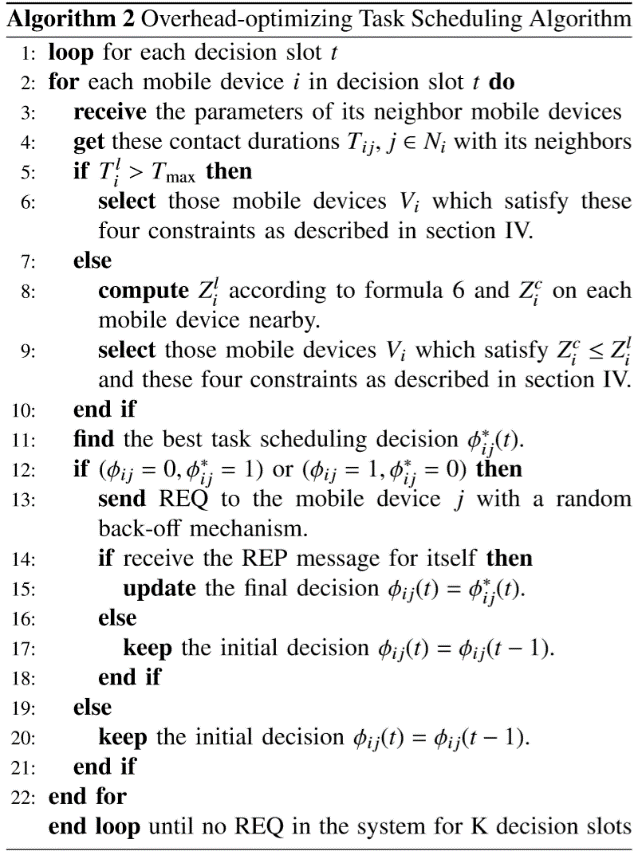
如果一个移动设备是一个为用户卸载有益的user，那么它将竞争决策更新的机会。如果移动设备得到机会，它可以更新其任务调度决策。否则，移动设备应再次进行分析，并竞争下一次更新机会。具体流程：

1）初始化。初始卸载决定设置为φij＝0。

2）运行任务调度机制。两次广播之间的时间为一个决策时隙。在一个决策时隙t中，能够使设备i的开销最小化的任务卸载决策φ\*ij（t）是**最优的调度决策**。在该时隙t中，由移动设备i和j在协商后做出的任务调度决策φij（t）是**最终调度决策**。

3）在t时，i的初始调度决策是φij（t-1）。如果i可更改此决策时隙中的调度策略来减少其开销，则定义更新响应BUi（t）。应用潜在博弈的有限改进特性，设计规则。

4）对于决策更新消息请求，采用random back-off mechanism，避免了来自不同移动设备的更新消息的冲突问题。



**算法分析**：计算复杂度为O（CN），the algorithm can be terminated within at most

iteration times.

**五、实验**

**对比算法：**

1）无任务调度的本地计算：每个移动设备选择自己执行任务

2）随机选择的设备进行任务调度

3）基于交叉熵的优化方案，采用集中式交叉熵方法求解任务调度问题

**指标：**

1. System performance
2. Influence factors analysis

(1) Communication data size

(2) Computation size