

Price-Based Distributed Offloading for Mobile-Edge Computing with Computation Capacity Constraints

Mengyu Liu and Yuan Liu, Member, IEEE

目标：边缘云设置价格使其**收益最大化**，每个用户单独设计卸载决策，以将其**成本**（延迟+支付）**最小化**。

系统模型：

K 个用户和一个基站

用户的计算数据可以按**位**任意划分，分为部分本地计算和部分卸载。

总带宽 B（平均分配给 K 个用户）

C_k : user k 计算 1 bit 的输入数据的 CPU cycles (cycles/bit)

R_k : user k 的输入数据大小 (bit)

l_k : 卸载到边缘云的数据大小 (bit)

$R_k - l_k$: 本地计算数据大小 (bit)

F_k : 用户 k 的本地 CPU 频率 (cycles/s)

本地计算时间: $t_{loc,k} = (R_k - l_k)C_k/F_k$

卸载时间: $t_{off,k} = t_{u,k} + t_{c,k} + t_{d,k}$.

$t_{\{off,k\}}$ = the uplink transmission time ($t_{\{u,k\}}$) + the execution time at the cloud $t_{\{c,k\}}$ + the downlink feedback time $t_{\{d,k\}}$

其中: $t_{u,k} = l_k/r_k$ $t_{d,k} = \alpha_k l_k/r_{B,k}$ $t_{c,k} = l_k C_k/f_{c,k}$ $f_{c,k} = f_C/K$

α_k 表示向 cloud 中加载的输出与输入比特的比例

$f_{\{c,k\}}$ 表示分配给用户 k 的边缘云的计算速度

f_C : cloud 的总的计算速度

总的时间: $t_k = \max\{t_{loc,k}, t_{off,k}\}$

边缘云有计算能力限制: 其计算每个加载周期中接收数据之和的 CPU 周期上限为 $F \sum_{k=1}^K \ell_k C_k \leq \bar{F}$.

Stackelberg Game Formulation:

应用 Stackelberg 博弈来模拟边缘云和用户之间的交互，其中边缘云是 leader，用户是 follower。Edge cloud (leader) 首先对用户的 CPU 周期定价。然后，用户 (follower) 根据 edge cloud 公布的价格，将输入数据分别划分为本地计算和卸载。

定义用户的 CPU 周期价格: $\mu = \{\mu_1, \dots, \mu_K\}$.

边缘云方面: 目标是最大化它的收益。优化问题表示为:

$$\mathbf{P1}: \max_{\mu \geq 0} U_B(\mu) = \sum_{k=1}^K \mu_k \ell_k C_k \quad \text{s.t.}$$

用户方面: 每个用户的成本=延迟时间+边缘云收取的费用: $U_k(\ell_k, \mu_k) = t_k + \mu_k \ell_k C_k$

$$U_k(\ell_k, \mu_k) = \begin{cases} (\mu_k - \frac{1}{F_k})\ell_k C_k + \frac{R_k C_k}{F_k}, & 0 \leq \ell_k \leq m_k, \\ \beta_k \ell_k + \mu_k \ell_k C_k, & m_k < \ell_k \leq R_k, \end{cases} \quad \text{where } \beta_k = \frac{1}{r_k} + \frac{C_k}{f_{c,k}} + \frac{\alpha_k}{r_{B,k}} \text{ and } 0 < m_k < R_k \text{ is defined as } m_k = \frac{C_k R_k}{\beta_k F_k + C_k}. \quad (4)$$

β_k : 1 bit 的数据卸载时间。 m_k 是一个分界线: $l_k \leq m_k$, 本地计算时间 > 卸载时间, $l_k > m_k$, 卸载时间 > 本地计算时间。

每个用户 k 的目标是根据由边缘云给定的价格 μ_k 选择最优卸载数据大小 l_k 来最小化其自身成本。问题表示为:

$$\mathbf{P2}: \min_{\ell_k} U_k(\ell_k, \mu_k) \quad \text{s.t.} \quad 0 \leq \ell_k \leq R_k$$

OPTIMAL ALGORITHM :

反向归纳：根据给定的价格 μ 通过解决 P2 得到 user k 的 卸载策略 l_k^* ；边缘云知道每个用户的卸载决策 $l_k^*(\mu_k)$ ，然后通过解决 P1 来确定其最优价格 μ^* 。

本文考虑了两种最优定价策略，即统一定价策略和差异化定价策略。

A. Uniform Pricing

$\mu = \mu_1 = \dots = \mu_K$ ，则目标函数 U_k 是 l_k 的线性分段函数，

Problem P2 的最优解为：

$$l_k^*(\mu) = m_k x_k, \quad \forall k, \quad (5) \quad x_k = \begin{cases} 1, & \mu \leq \frac{1}{F_k}, \\ 0, & \text{otherwise.} \end{cases}$$

如果用户 k 的 CPU 频率 $F_k \leq 1/\mu$ ，则 user k 倾向于将 m_k bit 数据加载到边缘云，否则将所有 bits 留给本地计算。

P1 改写为 P3:

$$\begin{aligned} \text{P3: } \max_{\mu \geq 0} \quad & U_B(\mu) = \mu \sum_{k=1}^K m_k x_k C_k \\ \text{s.t. } \quad & \sum_{k=1}^K m_k x_k C_k \leq \bar{F}. \end{aligned}$$

Algorithm 1 Optimal Uniform Pricing Policy for Problem P3

```
1: The edge cloud initializes  $\tau = K$  and  $\mu^\tau = 1/F_\tau$ .
2: repeat
3:   Every user decides its optimal offloaded data size  $l_k^*(\mu^\tau)$  according to (5).
4:   The edge cloud computes its revenue  $U_B(\mu^\tau)$  from (7).
5:   if  $\sum_{k=1}^K l_k^*(\mu^\tau) C_k \leq \bar{F}$  then
6:     Update the price  $\mu^{\tau-1} = 1/F_{\tau-1}$ , and  $\tau \leftarrow \tau - 1$ ;
7:   else
8:     Set  $U_B(\mu^\tau) = 0$ ; break;
9:   end if
10: until  $\tau \leq 0$ .
11: Output  $\mu^* \leftarrow \arg\max_{\mu^\tau} U_B(\mu^\tau)$ .
```

B. Differentiated Pricing

Problem P1 can be rewritten as : **P4:** $\max_{\mu \geq 0} U_B(\mu) = \sum_{k=1}^K \mu_k \boxed{m_k x_k} C_k$ s.t.

$x_k = 1$, $\mu_k^* = 1/F_k$ 。

$x_k = 0$, $\mu_k^* = \infty$ and earns no revenue.

P4 相当于 P4' : **P4':** $\max_{x_k \in \{0,1\}} U_B(x_k) = \sum_{k=1}^K \frac{m_k x_k C_k}{F_k}$ s.t.

P4' 实际上是一个二进制背包问题。应用动态规划[15]解决上述二元背包问题。

评估：

实验参数：

$B = 1$ MHz, $N_0 = -174$ dBm/Hz, h_k in $[-50, -30]$ dBm, F_k selected from the set $\{0.1, 0.2, \dots, 1\}$ GHz, $C_k \in [500, 1500]$ cycles/bit, $R_k \in [100, 500]$ KB, $p_k = 0.1$ W, $P_{\{B,k\}} = 1$ W, $F = 6 \times 10^9$ cycles/slot, $\alpha_k = 0.2$, $f_C = 100$ GHz.

实验结果：

比较：两种定价方案 + 所有输入数据都在用户处进行本地计算的方案

指标：计算容量对平均延迟和收入的影响

用户数量对平均延迟和收入的影响

附：

Stackelberg 博弈（斯塔克伯格博弈）的基本思想：

Stackelberg Game 是一个两阶段的完全信息动态博弈，博弈的 time 是序贯的。

主要思想是双方都是根据对方可能的策略来选择自己的策略以保证自己在对方策略下的利益最大化，从而达到纳什均衡。在该博弈模型中，先作出决策的一方被称为 leader，在 leader 之后，剩余的 players 根据 leader 的决策进行决策，被称为 followers，然后 leader 再根据 followers 的决策对自己的决策进行调整，如此往复，直到达到纳什均衡。

https://en.wikipedia.org/wiki/Stackelberg_competition