复现 5G 网络中移动数据卸载的联合在线边缘缓存和负载均衡

彭伟轩

摘要

本文复现的论文研究了 5G 网络中移动数据卸载的边缘缓存和负载均衡联合优化问题。首先提出了一种具有保证最优性的整数约束离线求解问题的原始对偶分解算法。然后论文将几种在线优化算法与提出的原始对偶算法相结合，在信息有限的情况下来进行决策解决在线问题。

关键词： 优化；原始对偶分解；在线算法

1 引言

随着智能边缘设备如智能手机和平板电脑的迅猛普及，社交媒体应用和视频直播等服务也得到广泛应用，从而极大地丰富了移动用户 (mobile user, MU) 的体验。为了满足移动用户对高速和低延迟的需求，无线通信网络必须相应地增加容量 [1]。为了实现这一目标，5G 无线网络在设计时采用了更大的带宽、更多的天线、更高的频率复用和网络致密化等关键技术 [2]。除了技术层面的改进，部署越来越多的新型小型基站 (small base stations, SBSs)，如微蜂窝、皮蜂窝和飞蜂窝基站 [3]，也成为了降低成本、提升服务质量的重要手段。这些小型基站通过反向链路与基站 (base station, BS) 连接，而移动用户可以根据需要选择与附近的 SBS 进行通信，而非总是与 BS 进行通信。由于距离较近，频谱复用和能耗也得以降低，因此用户与相邻 SBS 之间的传输成本要远低于 BS 之间的传输成本。

在提升服务质量的目标导向下，5G 技术引入了边缘缓存这一概念。在此架构中，5G 网络中的 SBS 配备移动计算服务器，不仅为缓存提供了必要的内存空间，还为智能决策提供了计算支持。随着内存设备成本的持续降低，为每个 SBS 配置有限但数量可观的内存空间以支持边缘缓存已成为现实。此外，网络流量的时间可变性为在流量较低的时段进行缓存更新提供了契机，从而有效降低了峰值流量需求和传输延迟 [4]。而计算设备的引入，使得 SBS 能够在众多用户请求之间实现精细的负载平衡，进一步提升服务的整体效能。

2 相关工作

2.1 边缘缓存

前人对蜂窝网络 (包括 5G 网络) 中的边缘缓存进行了大量研究 [5, 6]。其中内容流行度或 MU 的请求模式通常被建模为 Zipf 分布。缓存策略在 [6] 中被建模为整数规划。上述研究

大多未考虑系统更新成本等与时间相关的调整成本。然而，它已经在许多领域得到了广泛的考虑，例如云无线接入网 (C-RAN) 中的边缘缓存 [7] 和负载平衡 [8]。

2.2 在线凸优化

在线凸优化 (Online Convex Optimization, OCO) 有着悠久而丰富的历史，在计算机科学和其他领域有着广泛的应用。近年来，OCO 在网络和分布式系统社区的应用中引起了相当大的兴趣。特别是，OCO 为数据中心的动态容量规划、负载转移和需求响应 [9] 等情景实现了新颖的设计。

许多在线控制算法被用来解决考虑了时间相关调整成本的影响的 OCO 问题。滚动时域控制 (Receding Horizon Control, RHC)[10] 使用提前多步预测直接优化下一步行动。相比之下，平均固定水平控制 (Averaging Fixed Horizon Control, AFHC)[8] 通过平均多个固定视界控制 (Fixed Horizon Control, FHC) 算法的动作来工作，每个算法的工作原理与 RHC 相似，但在给定的预测视界内承诺所有 w 个动作。[11] 通过提供一类称为承诺地平线控制 (CHC) 的算法来推广 RHC 的工作。

3 本文方法

3.1 模型建立

考虑如图1所示的下载模型。一个区域有一个 BS 和 N 个 SBS，每个 SBS 覆盖的区域没有交集。每个 MU（表示为 m_n）从 BS 中下载特定文件 k(此文件可以被缓存在 SBS 中，MU 可以从其范围内的 SBS 中下载该文件)，下载速度为 $\lambda^t_{m_n,k}$。

决策变量:

- 使用 $x^t_{n,k} \in \{0,1\}$ 表示文件 k 是否在时刻 t 缓存在 SBS n 中。
- 使用 $y^t_{m_n,k} \in [0,1]$ 表示每个 MN 对在 t 时间由 SBS n 提供对内容 k 的请求服务的百分比。类似地，使用 $1 - y^t_{m_n,k}$ 表示每个 MN 对在 t 时间由 BS 提供对内容 k 的请求服务的百分比。

表1总结了重要的符号。

表 1. 问题表述中使用的符号表

Notation	Definition
$\mathcal{N}$	Set of SBS $\mathcal{N} = \{1, 2, \ldots, N\}$
$\mathcal{K}$	Set of files $\mathcal{K} = \{1, 2, \ldots, K\}$
M_n	Set of classes of MUs $M_n = \{m_n : n \in N\}$
T	Set of timeslots $T = \{1, 2, \ldots, T\}$
Λ_t	MUs requests matrix
$\lambda^t_{m_n,k}$	Demand of m_n for content k at time slot t
X^t	Set of caching variables $X_t = (x^t_{n,k})_{n\in N,k\in K}$
Y^t	Set of load balancing variables of SBS $Y_t = (y^t_{m_n,k})_{m_n\in M_n,k\in K}$
C_n	Cache size of SBS n
B_n	Bandwidth capacity of SBS n
ω_{m_n}	Weighted transmission parameter to BS of the classes MUs m_n
$\hat{\omega}_{m_n}$	Weighted transmission parameter to SBS n of the classes MUs m_n
$\mathbf{X^t}$	Vector of X^t
$\mathbf{Y^t}$	Vector of Y^t
β_n	Cache replacement parameter of SBS n
$z^t_{m_n,k}$	Load balancing variable of BS

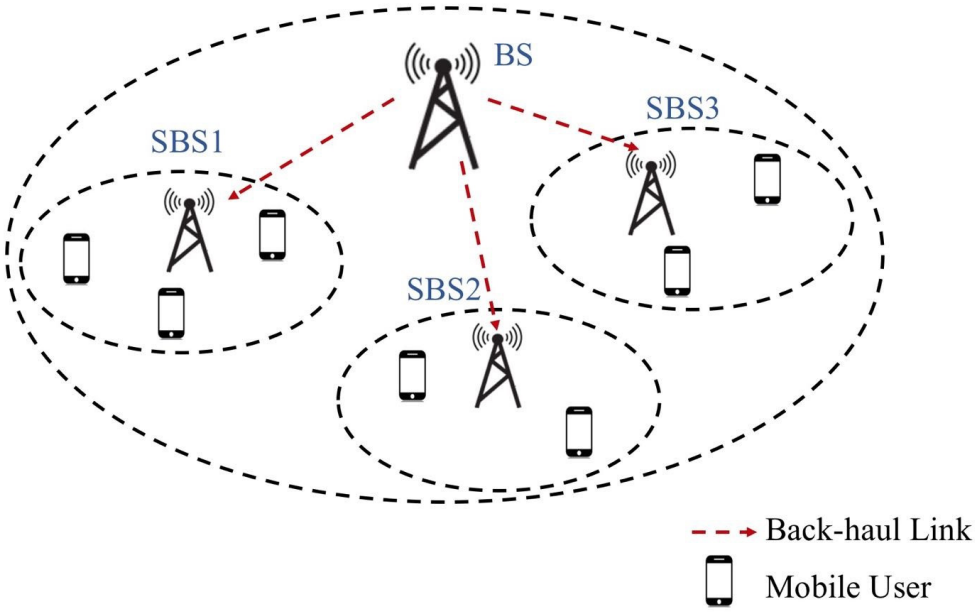

图 1. 论文提出的系统模型的一个例子。

3.2 状态方程

定义从 BS 为 MUs 提供服务的运营成本为：(1)

$$f_t(Y^t) = \sum_{n\in\mathcal{N}} \left(\sum_{m_n\in\mathcal{M}_n} \omega_{m_n} \sum_{k\in\mathcal{K}} (1 - y^t_{m_n,k})\lambda^t_{m_n,k} \right)^2. \tag{1}$$

定义从 SBS 为 MUs 提供服务的运营成本为：(2)

$$g_t(Y^t) = \sum_{n\in\mathcal{N}} \left(\sum_{m_n\in\mathcal{M}_n} \hat{\omega}_{m_n} \sum_{k\in\mathcal{K}} y^t_{m_n,k}\lambda^t_{m_n,k} \right)^2. \tag{2}$$

定义缓存替换成本为：(4).

$$d(x^t_n, x^{t-1}_n) = \beta_n \sum_k \left(x^t_{n,k} - x^{t-1}_{n,k}\right)^+, \tag{3}$$

$$h(X^t, X^{t-1}) = \sum_{n\in\mathcal{N}} \beta_n \sum_k \left(x^t_{n,k} - x^{t-1}_{n,k}\right)^+. \tag{4}$$

3.3 优化问题

随后论文提出的优化问题旨在通过在时间范围内为每个 SBS 和 MU 选择缓存策略 $x^t_{n,k}$ 和负载平衡策略 $y^t_{m_n,k}$ 来最小化由前面提到的三个成本组成的目标函数。优化问题公式如下：

$$\min_{\mathrm{X,Y}} \sum_{t\in\mathcal{T}^t} (f(Y^t) + g(Y^t) + h(X^t,\ X^{t-1})), \tag{5}$$

$$s.t. \quad x^t_{n,k} \in \{0,1\}, \forall n \in N, k \in \mathcal{K}, t \in \mathcal{T} \tag{6}$$

$$0 \le y^t_{m_n,k} \le 1, \forall n \in \mathcal{N}, m \in \mathcal{M}, k \in \mathcal{K}, t \in \mathcal{T}, \tag{7}$$

$$\sum_{k\in\mathcal{K}} x^t_{n,k} \le C_n, \forall n \in \mathcal{N}, t \in \mathcal{T}, \tag{8}$$

$$\sum_{k\in\mathcal{K}} \sum_{m_n\in\mathcal{M}_n} \lambda^t_{m_n,k} y^t_{m_n,k} \le B_n,\ \forall n \in \mathcal{N}, t \in \mathcal{T}, \tag{9}$$

$$y^t_{m_n,k} \le x^t_{n,k}, \forall n \in \mathcal{N}, m_n \in \mathcal{M}_n, k \in \mathcal{K}, t \in \mathcal{T}. \tag{10}$$

3.4 离线算法设计

由于上一节的问题是一个混合的整数规划问题，所以直接解决问题效率不高。论文提出了一种基于原始对偶分解方法来解决这个离线问题。该算法可进一步用于在线优化。

引入对偶拉格朗日乘子集来松弛 x 与 y 的耦合(10)：

$$\mu^t = (\mu^t_{n,m_n,k} \ge 0 : \forall n \in \mathcal{N},\ m_n \in \mathcal{M}_n,\ k \in \mathcal{K},\ t \in \mathcal{T}). \tag{11}$$

此时，对偶优化问题可以写成：

$$\begin{aligned}\max_{\mu^t} \min_{\mathrm{X}^t,\mathrm{Y}^t} L(\mathrm{X}^t, \mathrm{Y}^t,\ \mu^t) = &\sum_{t\in\mathcal{T}} (f(Y^t) + g(Y^t) + h(X^t,\ X^{t-1})) \\ &+ \sum_{n\in\mathcal{N}_n} \sum_{m\in\mathcal{M}_n} \sum_{k\in\mathcal{K}} \sum_{t\in\mathcal{T}} \mu^t_{n,m_n,k}(y^t_{m_n,k} - x^t_{n,k}),\end{aligned} \tag{12}$$

$$s.t.\ (8), (9), (6), (7), (11).$$

文章提出解决这个问题的算法如 Algorithm 1所示。其中 P_1、P_2 分别为(12)中与 x、y 相关的项构成的子问题。

Algorithm 1 原始-对偶算法

Input: T, β_n, Λ_t, ω_{m_n}, $\hat{\omega}_{m_n}$, accuracy level $\varepsilon = 0.0001$, maximum number of iterations L
Output: $\mathbf{X^t}$, $\mathbf{Y^t}$
Set $\mu = 0$, the lower bound $LB = -\infty$, the upper bound $UB = +\infty$, and $l = 1$.
while $\frac{UB-LB}{UB} > \varepsilon$ and $l < L$ **do**
    Solve sub-problems P_1 for $x^t_{n,k}$ and P_2 for $y^t_{m_n,k}$ (in parallel)
    Set h as the optimal value of the primal problem
    if $h > LB$ **then**
        $LB = h$
    end if
    Update UB as the optimal value of (5)
    Update dual variables using (13)
    $l = l + 1$
end while

使用次梯度下降法来更新对偶变量。

$$\begin{aligned} \mu^{t,(l+1)}_{n,m_n,k} &= [\mu^{t,(l)}_{n,m_n,k} + \delta^{(l)} g^{t,(l)}_{n,m_n,k})]^+, \\ \delta^{(l)} &= \frac{1}{1+\alpha \cdot l}, \\ g^{t,(l)}_{n,m_n,k} &= y^{t,(l)}_{m_n,k} - x^{t,(l)}_{n,k}. \end{aligned} \tag{13}$$

3.5 在线算法设计

滚动时域控制 (RHC): 下面两式等号右侧为在 $t \in [\tau, \tau+\omega]$ 内(5)的解 (FHC), 取 $t = \tau$ 来实施。

$$x^{\tau}_{RHC,n,k} = X^{\tau}_{\tau}(x^t_{RHC,n,k},\ \lambda_{\tau+w|\tau}). \tag{14}$$

$$y^{\tau}_{RHC,m_n,k} = Y^{\tau}_{\tau}(y^t_{RHC,m_n,k},\ \lambda_{\tau+w|\tau}). \tag{15}$$

承诺时域控制 (CHC): 把多个 FHC 叠加起来取平均，示意图如图2所示。

$$x^t_{CHC,n,k} = \frac{1}{r}\sum_{v=0}^{r-1} x^{t,(v)}_{CHC,n,k}, \tag{16}$$

$$y^t_{CHC,m_n,k} = \frac{1}{r}\sum_{v=0}^{r-1} y^{t,(v)}_{CHC,m_n,k}. \tag{17}$$

4 复现细节

4.1 与已有开源代码对比

本文没有参考相关算法源代码，(除了作者提供的产生符合 Zipf 分布的请求 λ 的函数)。

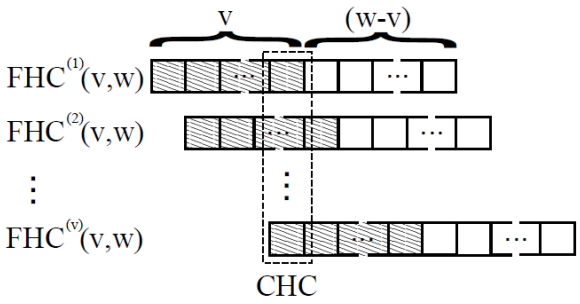

图 2. 承诺时域控制: 在每个时间步，它对 v FHC 算法定义的有限承诺的所有 v 动作进行平均。

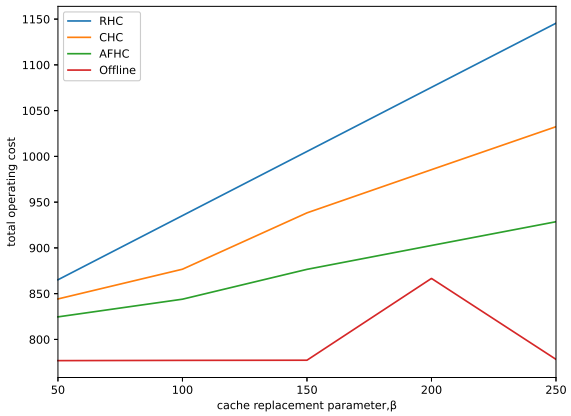

图 3. 总操作成本 vs 缓存替换成本 β.

4.2 实验环境搭建

本实验在 Python 3.12.0, scipy 1.11.4, numpy 1.26.1 版本下进行

4.3 使用说明

更换 const.py 文件中的参数，然后运行 main.py 即可。

5 实验结果分析

我执行了论文中的几个模拟实验来评估所提出算法的性能以及与时间相关的缓存替换成本的影响。

在线算法的性能：图3显示了各个算法的总成本，由于 AFHC 利用了更多的信息而不是只提交了第一项所以 AFHC 的性能优于 RHC。可以看出 CHC 的性能介于 RHC 和 AFHC 之间，由于 AFHC 是 CHC 的特例，所以 CHC 不会比 AFHC 好。

缓存替换成本 β 的影响：同样在图3中我们可以看到当缓存替换参数 β 增大时，离线算法和在线算法的运行成本也增大。但是离线算法在 $\beta = 200$ 时有突然增加可能是随机误差导致。

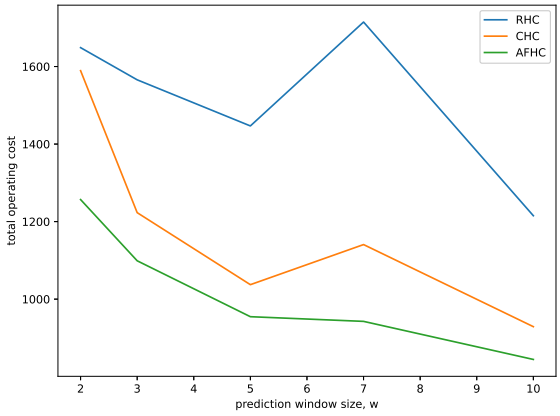

图 4. 总操作成本 vs 预测窗口大小 ω.

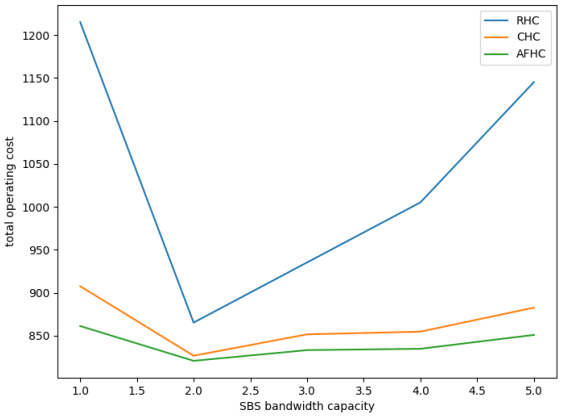

图 5. 总操作成本 vs SBS 带宽.

预测窗口 ω 对在线算法的影响：图4比较了 RHC、AFHC 和 CHC 在预测窗口大小 w 变化时的总运行成本变化情况。注意到随着预测窗口的增大，所有在线算法都有下降趋势。我们可以得出结论，当系统具有更多的 MUs 请求预测信息时，在线算法的性能更好。

SBS 带宽容量的影响：图5描述了当 SBS 的带宽容量发生变化时，总操作成本的变化情况。当 SBS 的带宽容量变大时，意味着 SBS 可以在每个时隙发送更多的项目来满足 MU 的请求，所以在 bandwidth > 1 时的成本会比带宽容量为 1 时的成本显著降低。当 bandwidth>1 之后增大带宽会使成本不变甚至升高，原因可能是在 bandwidth=2 时已经可以满足所有 MU 的请求，再增大窗口无法进一步降低总成本。

6 总结与展望

本文复现了 5G 网络中移动数据卸载的边缘缓存和负载均衡联合优化问题。实现了基于原始对偶分解的算法，并与几种在线优化算法相结合。实验结果大致还原了论文中的结果。然而由于我使用了和文章不同的优化器，我的复现结果和论文有一些出入，而且有一些趋势之外的噪点，这是我实现工作中的不足之处。未来我会使用其它的优化器如 cvxpy 进行求解。

References

[1] Sean Kenneth Barker and Prashant Shenoy. Empirical evaluation of latency-sensitive application performance in the cloud. In *Acm Sigmm Conference on Multimedia Systems*, 2010.

[2] Akhil Gupta and R. K. Jha. A survey of 5g network: Architecture and emerging technologies. *IEEE Access*, 3:1206–1232, 2015.

[3] Amitabha Ghosh, Nitin Mangalvedhe, Rapeepat Ratasuk, Bishwarup Mondal, Mark Cudak, Eugene Visotsky, Timothy A. Thomas, Jeffrey G. Andrews, Ping Xia, Han Shin Jo, Harpreet S. Dhillon, and Thomas D. Novlan. Heterogeneous cellular networks: From theory to practice. *IEEE Communications Magazine*, 50(6):54–64, 2012.

[4] Mohammad Ali Maddah-Ali and Urs Niesen. Fundamental limits of caching. In *2013 IEEE International Symposium on Information Theory*, pages 1077–1081, 2013.

[5] Konstantinos Poularakis, George Iosifidis, and Leandros Tassiulas. A framework for mobile data offloading to leased cache-endowed small cell networks. In *2014 IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems*, pages 327–335, 2014.

[6] Jun Li, Youjia Chen, Zihuai Lin, Wen Chen, Branka Vucetic, and Lajos Hanzo. Distributed caching for data dissemination in the downlink of heterogeneous networks. *IEEE Transactions on Communications*, 63(10):1–1, 2015.

[7] Lingjun Pu, Lei Jiao, Xu Chen, Lin Wang, Qinyi Xie, and Jingdong Xu. Online resource allocation, content placement and request routing for cost-efficient edge caching in cloud radio access networks. *IEEE Journal on Selected Areas in Communications*, 36(8):1751–1767, 2018.

[8] Minghong Lin, Zhenhua Liu, Adam Wierman, and Lachlan L. H. Andrew. Online algorithms for geographical load balancing. In *International Green Computing Conference*, 2012.

[9] Zhenhua Liu A, Adam Wierman A, Yuan Chen B, Benjamin Razon A, and Niangjun Chen A. Data center demand response: Avoiding the coincident peak via workload shifting and local generation. *Performance Evaluation*, 70(10):770–791, 2013.

[10] H. Michalska and D.Q. Mayne. Robust receding horizon control of constrained nonlinear systems. *IEEE Transactions on Automatic Control*, 38(11):1623–1633, 1993.

[11] Joshua, Comden, Zhenhua, Liu, Anshul, Gandhi, Adam, Wiermar, Niangjun, and Cher. Using predictions in online optimization: Looking forward with an eye on the past. *Performance Evaluation Review*, 44(1):193–206, 2016.