基于博弈论的基站间缓存分配技术研究

姚乐，范文浩，刘元安，吴帆

（北京邮电大学电子工程学院，北京 100876）

摘要：移动边缘计算环境中的智能基站网络一般由多个配备移动边缘计算服务器的基站（MEC-BS）构成。如何充分利用 MEC-BS 的有限存储能力，提高流行内容缓存文件数量是该领域的核心技术问题。本文提出了一种最大化效用的基于博弈论的基站间缓存分配技术。该技术考虑了网络服务提供商（NSP）、流行内容提供商（PCPs）和 MEC-BSs 组合的异构网络，根据用户对流行内容文件的需求和 MEC-BS 存储约束建立了缓存模型，分别建立了异构网络中不同的利润模型，并依据该模型将所研究问题定义为一个斯塔克伯格博弈。最后设计一种异步算法模拟非合作博弈，利用仿真分析所提算法的快速收敛性以及领导者效应函数凹性的理论正确性。

关键词：移动边缘计算；缓存分配；博弈论；异步算法

中图分类号： TP393

Game Theoretic Research for Cache Allocation among Base Stations

YAO Le，FAN Wenhao，LIU Yuan'an，WU Fan

(School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100876)

Abstract: The intelligent base station network in mobile edge computing environment generally consists of several base stations (MEC-BS) equipped with mobile edge computing server. How to make full use of the limited storage capacity of MEC-BS and improve the quantity of popular content files is the core technical problem in this field. In this paper, a game theory-based cache allocation technique for maximizing utility among MEC-BSs is proposed. This technology considers a heterogeneous network consisting of network service providers (NSP), popular content providers (PCPs), and MEC-BSs. A caching model is actually established which is based on the users' demonds for popular content files and the storage constraints of MEC-BS. Different profit models are built in this heterogeneous network and the research problem is defined as a Starkberg game. Finally, an asynchronous algorithm is designed to simulate the non-cooperative game, and the fast convergence of the proposed algorithm and the theoretical correctness of the concavity of the leader's utility function are analyzed by simulation.

基金项目：国家自然科学基金 (61821001)，广东省扬帆计划项目

作者简介：姚乐（1995-），男，硕士研究生，主要研究方向：边缘计算。通信作者：刘元安（1963-），男，教授，主要研究方向：电磁兼容，普适计算和无线通信. E-mail: yuliu@bupt.edu.cn

Key words: Mobile Edge Computing;Cache Allocation;Game Theory;Asynchronous Algorithm

0 引言

移动边缘计算（Mobile Edge Computing，MEC）作为云计算向网络边缘的扩展，已经成为 5G 通信网络中的核心技术之一[1]。在已有的诸多研究中，配备有 MEC 服务器的基站被称为移动边缘计算智能基站（Mobile Edge Computing-Base Station，MEC-BS），作为靠近用户侧的边缘节点，已经可以满足用户很多要求，如低时延、高密度等。但是，随着 5G 通信网络的发展，更好的网络质量又受到了流行内容提供商（Popular Content Provider，PCP）的关注。流行内容提供商提供的流行内容（如媒体资源）数量逐渐增多，并且数据量小。这就带来一个问题，如果所有的流行内容都使用回程链路连接至核心网，再进行资源的下载，那么必将带来高额的成本。为了解决这个问题，基站中已经配备了具有存储能力的 MEC 服务器，但是由于容量的限制，单 MEC-BS 还是无法有效的应对，因此基于基站间协作的的缓存分配技术便成了解决该问题的有力方法。

博弈论是研究具有竞争性质或者斗争现象的数学理论。将博弈论应用于解决 MEC-BS 之间的优化问题，是该领域的又一热点[2, 3, 4]。博弈论的运用有效地解决了流行内容在 MEC-BS 之间的分配问题，使所有博弈中的玩家达到纳什均衡。

本文提出了一种基于博弈论的基站间缓存分配技术。考虑网络服务提供商（Network Service Provider，NSP）和流行内容提供商的异构网络，建立基于利润模型的网络服务提供商和流行内容提供商的优化模型，构建了玩家地位不均等的斯塔克伯格博弈，理论求解出领导者和跟随者的最佳方案，并设计异步算法模拟博弈，通过仿真分析算法性能和理论分析的正确性。

1 系统模型

1.1 网络模型

我们考虑一个异构网络（Heterogeneous Networks，HetNets），由一个 NSP、M 个 MEC-BSs 和 N 个 PCPs 组成。用集合 $\mathcal{M}$ 表示具备缓存能力的 MEC-BSs，即 $\mathcal{M} = \{1, \cdots, M\}$，其中每个 MEC-BS 都有一个有限存储容量 s_m。另外，集合 $\mathcal{N} = \{1, \cdots, N\}$ 包含了所有 PCPs。

每一个流行内容提供商都有自己的本地流行内容目录，假定将其表示为若干流行文件，表示为 F_n。那么全局流行文件目录则可以用集合 $\mathcal{F} = \{f_1, \cdots, f_N\}$ 表示，每一个用户都可以向基站请求自己感兴趣的流行文件。现如今，文件编码技术非常成熟，已有许多研究将流行文件考虑为同等大小[5, 6]，因此假设所有的文件具有相同的单元大小 s。每个流行内容提供商的目标是在基站本地缓存其最受欢迎的文件副本，以便使用户体验到更高的数据传输速率。但是，由于基站的存储容量有限，并且并非所有文件都可以缓存有足够的副本。

1.2 缓存模型

假设用户对流行文件的请求概率已知，对于流行内容提供商来说，它含有若干可以缓存在基站的流行文件，但是由于存储能力的限制，流行内容提供商需要有选择的对流行文件缓存。因此，我们定义 q_n 表示流行内容提供商 n 缓存的流行文件排序，具体来说，如果将该流行内容提供商所能缓存的流行文件按照流行程度（请求概率）逆序排列，则 q_n 表示其可以缓存前 q_n 个流行文件。将所有流行内容提供商的缓存文件写进一个集合 $\boldsymbol{q}=\{q_1,\cdots,q_N\}$。那么，相应地其必须满足以下存储能力的约束

$$\sum_{n=1}^{N}\sum_{1}^{q_n} s \le s_m, \quad f_{q_n}\in\mathcal{F} \tag{1}$$

1.3 利润模型

1.3.1 NSP 利润模型

NSP 作为整个系统唯一的管理者，掌控着所有的资源（存储、网络等），对其中所存在的所有其他服务商收取费用是其唯一的收入来源（类比于房东向租户收取租金）。如果 PCP 需要借助在 NSP 管理的基站来缓存流行文件，那么便需要提供一定的费用，这里的费用由 NSP 来实时提供。为了保证对 PCP 提供公平的缓存服务，我们定义缓存一个流行文件的价格为 φ。在第 1.2节中已经定义缓存模型，即 PCP n 请求缓存的流行文件数为 q_n，那么 NSP 的收入可以写为

$$R_0=\varphi\sum_{n=1}^{N} q_n \tag{2}$$

NSP 做为管理者其成本也很清楚，那就是维护成本。当缓存在系统中的文件数越多，那维护成本越高，我们假设该成本与缓存文件数量成正比，用 C_0 表示成本，那么其具体表达式可以写为

$$C_0=\frac{1}{S-S(\boldsymbol{q})} \tag{3}$$

公式（3）中，S 表示系统内总的存储容量，其可以写成 $S=\sum_{m=1}^{M} s_m$，$S(\boldsymbol{q})$ 表示所有 PCPs 缓存文件的总存储容量，具体形式为

$$S(\boldsymbol{q})=s\sum_{n=1}^{N} q_n \tag{4}$$

另外，公式（3）必须满足 $S-S(\boldsymbol{q})>0$ 的条件，否则我们认为成本无穷大（$C_0=\infty$）。

现在，我们可以得到 NSP 的利润模型，它是关于所有 PCPs 缓存文件数量的函数，即

$$u_0=R_0-C_0=\varphi\sum_{n=1}^{N} q_n-\frac{1}{S-S(\boldsymbol{q})} \tag{5}$$

可以看出，NSP 的成本由流行文件的价格 φ 和所有 PCPs 缓存文件数量 $\boldsymbol{q}$ 共同决定。

1.3.2 PCP 利润模型

对于 PCP 来说，其主要目的就是能够尽可能的在 MEC-BS 中缓存自己的流行文件，这样可以最大限度的减少回程链路的使用，将用户处的时延降低并提升其体验。在这个过程中，如果以指定的 PCP n 为例，一方面需要使得其缓存文件数量 q_n 越多越好，另一方面，假设用户对于 PCP n 的流行文件请求数为 η_n，那么该值也是越大越好，因为其代表了对于缓存文件的请求频率。上述两个参数均是正向参数，而对于系统内从全局角度来看，如果除了 PCP n 其余所有 PCPs 缓存的文件数越多，那么由于存储容量的限制，PCP n 便因 MEC-BS 没有足够的存储空间而减少缓存文件数量，进而降低用户体验。定义 $\boldsymbol{q}_{-n}$ 作为除了 PCP n 其余所有 PCPs 缓存文件数向量，因此，我们可以得出 PCP 收入同上述三种参数的关系表达式，即

$$R_n = \ln\left(1 + \frac{\alpha q_n}{1 + \frac{1}{\beta\eta_n}\gamma \sum \boldsymbol{q}_{-n}}\right) \tag{6}$$

从公式（6）中可以看出，PCP n 的收入同缓存文件数量 q_n 和用户对流行文件请求数 η_n 成正比，而对其他 PCP 缓存文件数 $\boldsymbol{q}_{-n}$ 成反比。这里，我们定义三个权重系数，分别是 α、β 和 γ，并且均为正数，式中权重的大小决定着哪一项参数更为重要。

由于在第 1.3.1节中已经说明，NSP 作为管理者承担所有缓存文件的维护成本，因此，所有 PCPs 只需支付在基站中缓存文件需要的费用即可。根据之前所定义的缓存一个流行文件的价格为 φ，那么对于 PCP n 来说，其所承担的成本（缓存文件费用）可表示为

$$C_n = \varphi q_n \tag{7}$$

那么，当给定 NSP 提出的缓存价格 φ 和其他 PCPs 缓存文件数 $\boldsymbol{q}_{-n}$ 之后，PCP n 的利润 u_n 可以写为

$$u_n = R_n - C_n = \ln\left(1 + \frac{\alpha q_n}{1 + \frac{1}{\beta\eta_n}\gamma \sum \boldsymbol{q}_{-n}}\right) - \varphi q_n \tag{8}$$

2 基站间缓存分配斯塔克伯格博弈

依据问题背景，NSP 作为领导者，它的目标是使得自身的利润最大化，而 PCPs 作为跟随者，它们的目标也是使得自身的利润最大化。在问题的描述中，不难发现，NSP 的利润受缓存文件价格和 PCPs 缓存文件数量共同影响，而 PCP 的利润同样如此。

我们将该问题描述为一个玩家博弈地位不均等的斯塔克伯格博弈，具体的博弈形式用公式表示为

$$\mathcal{G} = \left\langle \mathcal{I}(Leader), \mathcal{N}(Followers), \varphi, \{\mathcal{Q}_n\}_{n\in\mathcal{N}}, u_0, \{u_n\}_{n\in\mathcal{N}} \right\rangle \tag{9}$$

斯塔克伯格博弈（9）中，$\mathcal{I}$ 表示领导者（NSP），$\mathcal{N}$ 表示跟随者（所有的 PCPs），φ 表示领导者的策略，$\mathcal{Q}_n$ 表示跟随者的策略集，u_0 表示领导者的效用函数，u_n 表示跟随者的效用函数。这里为了方便起见，我们规定领导者的策略为缓存文件的价格 φ，跟随者的策略集为所有 PCPs 缓存文件数量集，领导者的效用函数为 NSP 的利润，跟随者的效用函数为 PCPs 的利润。

2.1 PCP 优化问题

2.1.1 PCP 最佳响应模型

当 NSP 给出缓存文件的价格 φ 之后，每个 PCP 会根据该价格开展一场玩家博弈地位均等的博弈，从最初的各自缓存最多文件的策略，由于存储容量的约束，会进行多次玩家之间的博弈，在每次博弈中，每个 PCP 均会给出自己的最佳响应，直至整体达到均衡。

我们以 PCP n 为例，给出其优化目标及相关约束如下

$$\text{maximize} \quad u_n(q_n) \qquad q_n \in \mathcal{Q} \tag{10}$$

$$\text{s.t.} \quad \sum_{m=1}^{M}\sum_{n=1}^{N}\sum_{1}^{q_n} s \le S, \quad f_{q_n} \in \mathcal{F} \tag{10a}$$

$$q_n \ge 0 \tag{10b}$$

约束（10a）满足了系统存储限制的条件，其中 S 的表达式在第 1.3.1节已经给出。根据目标函数（10），同样可以得到 PCP n 的缓存文件数，如下

$$q_n = \arg\max_{q_n \in \mathcal{Q}} u_n(q_n) \tag{11}$$

根据 PCP n 的优化目标及 q_n 的求解方式，我们给出 PCP n 最佳响应的理论推导。

首先，我们验证 PCP n 效用函数 u_n 的凹性。由于在计算最佳响应时，其他 PCPs 策略相对固定，所以求解效用函数 u_n 关于缓存文件数量 q_n 的一阶偏导如下

$$\frac{\partial u_n}{\partial q_n} = \frac{\alpha\beta\eta_n}{\alpha\beta q_n\eta_n + \beta\eta_n + \gamma\sum \boldsymbol{q}_{-n}} - \varphi \tag{12}$$

进而求解其二阶偏导如下

$$\frac{\partial^2 u_n}{\partial {q_n}^2} = \frac{-\alpha^2\beta^2\eta_n^2}{(\alpha\beta q_n\eta_n + \beta\eta_n + \gamma\sum \boldsymbol{q}_{-n})^2} \tag{13}$$

二阶偏导（13）中，分母是一个平方项，又因为其中每一项都大于 0，所以分母大于 0。同理，分子小于 0，进而使得 PCP n 效用函数 u_n 的二阶偏导 $\frac{\partial^2 u_n}{\partial {q_n}^2} < 0$。根据凹函数的定义与二阶偏导的关系，当函数的二阶偏导存在且小于 0，则该函数是一个凹函数。那么，当其他 PCPs 策略固定时，PCP n 的效用函数 u_n 存在最大值，且唯一。因此，我们给出 PCP n 的最佳响应策略如下

$$q_n^{\circ} = \frac{1}{\varphi} - \frac{1}{\alpha} - \frac{\gamma\sum \boldsymbol{q}_{-n}}{\alpha\beta\eta_n} \tag{14}$$

2.1.2 PCP 纳什均衡

在第 2.1.1节中，我们定义了 PCP 最佳响应模型，它存在的前提是各个 PCP 在互相博弈，根据其他玩家的决策制定自己的最优决策，所以说其实局部考虑的。由于我们已经证明了 PCP n 的效用函数 u_n 是一个严格的凹函数，因此我们可以认为在全局存在一个最优的策略。不难证明，当效用函数 u_n 是一个严格的凹函数时，该非合作博弈会存在并且具有唯一地纳什均衡点。因此，我们给出该非合作博弈纳什均衡的定义，即

定义 1 (基站间缓存分配非合作博弈纳什均衡点). 我们定义一个迁移策略集 $\boldsymbol{q}^* = (q_1^*, \cdots, q_N^*)$，该集合表示所提出博弈的纳什均衡点。如果该均衡点 $\boldsymbol{q}^*$ 已经达到，那么没有任何一个 *PCP* 可以单方面调整其缓存策略从而增加效益。具体表示如下：

$$q_n^* = \arg\max_{q_n \in \mathcal{Q}} u_n(q_n, \boldsymbol{q}_{-n}^*) \tag{15}$$

式中 $\boldsymbol{q}_{-n}^* = \left\{q_1^*, \cdots, q_{n-1}^*, q_{n+1}^*, \cdots, q_N^*\right\}$ 表示在达到纳什均衡点时，除了 *PCP* n 之外其他所有 *PCPs* 的缓存策略集。基于策略集 $\boldsymbol{q}_{-n}^*$，*PCP* n 可以获得其最优缓存迁移策略集。

有了上述纳什均衡的定义之后，我们将所有 PCPs 的最佳响应策略联合起来写成一个多元线性方程组，具体形式如下所示

$$\begin{cases} q_1 + \dfrac{\gamma}{\alpha\beta\eta_1} \cdot q_2 + \cdots + \dfrac{\gamma}{\alpha\beta\eta_1} \cdot q_N = \dfrac{1}{\varphi} - \dfrac{1}{\alpha} \\ \dfrac{\gamma}{\alpha\beta\eta_2} \cdot q_1 + q_2 + \cdots + \dfrac{\gamma}{\alpha\beta\eta_2} \cdot q_N = \dfrac{1}{\varphi} - \dfrac{1}{\alpha} \\ \qquad\qquad\qquad\qquad\qquad\qquad \vdots \\ \dfrac{\gamma}{\alpha\beta\eta_N} \cdot q_1 + \dfrac{\gamma}{\alpha\beta\eta_N} \cdot q_2 + \cdots + q_N = \dfrac{1}{\varphi} - \dfrac{1}{\alpha} \end{cases} \tag{16}$$

从方程组（16）中可以看出，等式右边均为 $(\frac{1}{\varphi} - \frac{1}{\alpha})$，且该常数项大于 0，因此称该方程组为非齐次线性方程组。为了求解该方程组，我们引入克莱姆法则并利用矩阵的相关知识。首先我们定义一个系数矩阵，即

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{bmatrix} = \begin{bmatrix} 1 & \frac{\gamma}{\alpha\beta\eta_1} & \cdots & \frac{\gamma}{\alpha\beta\eta_1} \\ \frac{\gamma}{\alpha\beta\eta_2} & 1 & \cdots & \frac{\gamma}{\alpha\beta\eta_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\gamma}{\alpha\beta\eta_N} & \frac{\gamma}{\alpha\beta\eta_N} & \cdots & 1 \end{bmatrix} \tag{17}$$

然后，分别定义一个缓存策略组成的列向量和方程组右侧常数项组成的列向量，其维度均是 N 维，具体形式如下

$$Q = (q_1, q_2, \cdots, q_N)^T \tag{18}$$

和

$$B = (b_1, b_2, \cdots, b_N)^T = \underbrace{(\frac{1}{\varphi} - \frac{1}{\alpha}, \frac{1}{\varphi} - \frac{1}{\alpha}, \cdots, \frac{1}{\varphi} - \frac{1}{\alpha})^T}_{N} \tag{19}$$

联合公式（17）、（18）和（19）定义的三个参数，我们可以将方程组（16）写成如下矩阵形式 $AQ = B$。不难证明，系数矩阵 A 是一个非奇异矩阵。那么，利用克莱姆法则（Cramer's Rule），我们可以得到 PCP n 的纳什均衡点缓存策略，如下

$$q_n^* = \frac{\det(\mathrm{A_n})}{\det(\mathrm{A})} \tag{20}$$

式中 $\det(\mathrm{A})$ 表示系数矩阵 A 的行列式，$\det(\mathrm{A_n})$ 则表示将系数矩阵 A 的第 n 列换成常数列向量而其余列元素保持不变的行列式。

2.2 NSP 优化问题

假设我们已经通过一定方法预测到了所有 PCPs 的缓存决策 $\boldsymbol{q}$，那么对于 NSP 来说，它主要通过求解下面这个问题来优化自己提出的缓存文件价格 φ，即

$$\text{maximize} \quad u_0(\boldsymbol{q}) \qquad q_n \in \mathcal{Q} \tag{21}$$

$$\text{s.t.} \quad \sum_{m=1}^{M}\sum_{n=1}^{N}\sum_{1}^{q_n} s \le S, \quad f_{q_n} \in \mathcal{F} \tag{21a}$$

$$q_n \ge 0 \tag{21b}$$

上式中给出了 NSP 的目标函数，使得自身利润 u_0 最大化，那么关于最优缓存文件价格 φ 就可以写为

$$\varphi^* = \arg\max_{\varphi} u_0(\boldsymbol{q}) \tag{22}$$

在公式（5）中，我们已经定义了 NSP 的利润函数，如果我们使用流行内容提供商构建的非合作博弈的纳什均衡点作为对其缓存策略的预测，那么我们可以重新写利润 u_0 的表达式如下

$$\begin{aligned} u_0 &= \varphi\sum_{n=1}^{N} q_n - \frac{1}{S - S(\boldsymbol{q})} \\ &= \varphi\sum_{n=1}^{N} q_n^* - \frac{1}{S - s\sum_{n=1}^{N} q_n^*} \\ &= \varphi\sum_{n=1}^{N} (\frac{\det(\mathrm{A_n})}{\det(\mathrm{A})}) - \frac{1}{S - s\sum_{n=1}^{N} (\frac{\det(\mathrm{A_n})}{\det(\mathrm{A})})} \end{aligned} \tag{23}$$

式中矩阵 A_n 和矩阵 A 已在公式（20）中定义。这里以 A_1 为例，将其展开为

$$A_1 = \begin{bmatrix} \frac{1}{\varphi} - \frac{1}{\alpha} & a_{12} & \cdots & a_{1N} \\ \frac{1}{\varphi} - \frac{1}{\alpha} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\varphi} - \frac{1}{\alpha} & a_{N2} & \cdots & a_{NN} \end{bmatrix} = (\frac{1}{\varphi} - \frac{1}{\alpha}) \begin{bmatrix} 1 & a_{12} & \cdots & a_{1N} \\ 1 & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & a_{N2} & \cdots & a_{NN} \end{bmatrix} = (\frac{1}{\varphi} - \frac{1}{\alpha}){A_1}' \tag{24}$$

公式（24）中，令矩阵 $A_1 = (\frac{1}{\varphi} - \frac{1}{\alpha}){A_1}'$，其中的矩阵 ${A_1}'$，是将矩阵 A_1 的第一列全部提公因式之后剩下的矩阵，也就是第一列整体变成元素全为 1 的列向量。那么其一般形式可以写成 $A_N = (\frac{1}{\varphi} - \frac{1}{\alpha}){A_N}'$。然后我们定义 $T = \sum_{n=1}^{N} (\frac{\det(\mathrm{A_n}')}{\det(\mathrm{A})})$。有了上述公式的辅助，我们可以将公式（23）重写为

$$u_0 = \varphi(\frac{1}{\varphi} - \frac{1}{\alpha})T - \frac{1}{S - s(\frac{1}{\varphi} - \frac{1}{\alpha})T} \tag{25}$$

我们对 NSP 利润函数 u_0 求一阶偏导，如下

$$\frac{\partial u_0}{\partial \varphi} = -\frac{T}{\alpha} + \frac{SsT\alpha^2}{(S\alpha\varphi + sT\varphi - sT\alpha)^2} \tag{26}$$

进而得到二阶偏导，如下

$$\frac{\partial^2 u_0}{\partial \varphi^2} = \overbrace{-2SsT\alpha^2}^{\mathcal{Z}_1} \underbrace{(S\alpha + sT)}_{\mathcal{Z}_2} \frac{\overbrace{(S\alpha\varphi + sT\varphi - sT\alpha)}^{\mathcal{Z}_3}}{(S\alpha\varphi + sT\varphi - sT\alpha)^4} \tag{27}$$

根据凹函数定义，只有当我们的二阶偏导 $\frac{\partial^2 u_0}{\partial \varphi^2} < 0$ 时，利润函数 u_0 是一个凹函数，才具备最大值。我们仔细观察公式（27），$\mathcal{Z}_1$ 和 $\mathcal{Z}_2$ 恒大于 0，因此，我们只需满足 $\mathcal{Z}_3 > 0$ 即可。又因为公式（16）中已经说明 $(\frac{1}{\varphi} - \frac{1}{\alpha}) > 0$，综上两个条件，我们得出缓存文件价格 φ 的范围为

$$\frac{sT\alpha}{S\alpha + sT} < \varphi < \alpha \tag{28}$$

当缓存文件价格 φ 满足上述不等式后，利润函数 u_0 便是一个凹函数，那么利润函数 u_0 最大时对应的最优缓存文件价格 φ^* 为

$$\varphi^* = \frac{\sqrt{\frac{SsT\alpha^3}{T}} + sT\alpha}{S\alpha + sT} \tag{29}$$

3 基站间缓存分配斯塔克伯格博弈异步算法

在本节中，我们对本章所研究的基站间缓存分配斯塔克伯格博弈中流行内容提供商之间的非合作博弈提出了一种异步算法。在该算法中，每个流行内容提供商使用该算法与其他玩家进行博弈，在博弈中逐步获得最佳缓存策略。异步算法的主要描述如算法 1所示

算法 1 基站间缓存分配斯塔克伯格博弈异步算法

输入： 所有 MEC-BSs 集合 $\mathcal{M}$；所有流行内容提供商集合 $\mathcal{N}$；所有流行内容提供商初始策略集合 $\mathcal{Q}$；网络服务提供商提出缓存文件价格 φ；算法收敛因子 ϵ；

输出： 最优缓存策略集 $\boldsymbol{q}^*$;

1: **初始化：**选择任意合适的策略集 $\boldsymbol{q}^{(0)} \in \mathcal{Q}$，设置迭代计数器 $k = 0$ 并设置初始最优策略集 $\{\overline{q_n}\}_{n\in\mathcal{N}} = \mathbf{0}$;
2: **while** $\|\boldsymbol{q}^{(k)} - \boldsymbol{q}^{(k-1)}\| > \epsilon$ **do**
3: **for** $n = 1, \cdots, N$ **do**
4: $q_n^{(k+1)} = \arg\max_{q_n \in \mathcal{Q}} u_n(q_n) = \frac{1}{\varphi} - \frac{1}{\alpha} - \frac{\gamma \sum \boldsymbol{q}_{-n}}{\alpha\beta\eta_n}$;
5: 流行内容提供商 n 根据第 4步计算当前迭代的最优策略 $q_n^{(k+1)}$;
6: **end for**
7: $k \leftarrow k + 1$;
8: **end while**
9: **return** $\boldsymbol{q}^{(k)}$

上述算法的核心思想同样是使用迭代的思想模仿玩家博弈的过程，在每一次迭代中都会产生所有玩家这次博弈之后的临时最优策略集，但该算法更为靓点的地方在于第 3步到第 6步。该

阶段每个流行内容提供商可以异步的执行，并更新自己最新的最佳分配策略。最终，如果该异步算法达到理想的终止条件 $\|\boldsymbol{q}^{(k)}-\boldsymbol{q}^{(k-1)}\|<\epsilon$，则输出所有流行内容提供商的最佳缓存分配策略。

4 仿真及结果分析

在本节中，我们将通过仿真测试验证本章所提出的基站间缓存分配斯塔克伯格博弈异步算法的收敛性和理论正确性。为了能够真实模拟环境，我们设置基站的数量是 10 个，流行内容提供商的数量是 5 个，流行内容文件数是 40 个，每一个文件的大小在 50 ∼ 100Mb 之间，每一个流行内容提供商处用户对于流行文件的请求数范围在 20 ∼ 70 次。

4.1 算法收敛性分析

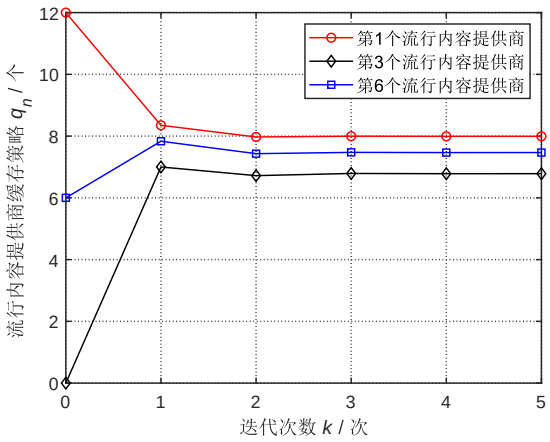

图 1: 异步算法收敛性

本章提出的算法 1是针对流行内容提供商之间非合作博弈的过程模拟，并且在第 2.1节中，我们已经对该非合作博弈进行了数学理论的证明，该纳什均衡存在并且唯一。为了通过仿真验证该理论，我们对算法 1进行数值分析，得到图 1。在图 1中可以看到，随着迭代次数 k 的增加，不同的流行内容提供商的缓存策略（缓存文件数）会逐渐趋于平稳，并且在 2 次迭代之后几乎不会变化，这就直接的验证了所提异步算法的快速收敛性。另外，观察三条曲线的起始点，第 3 个和第 6 个流行内容提供商的缓存策略从少到多（尤其是第 6 个，从 0 开始增加），而第 1 个流行内容提供商的缓存策略有所减少。这种现象也验证了所提算法会使得系统整体的缓存文件数变多，达到系统性能提升的效果。

4.2 网络服务提供商效用函数分析

通过本章的分析，我们知道影响网络服务提供商效用函数的因素有很多。在本节中，将对网络服务提供商效用函数 u_0 进行分析，分别是基站存储容量和缓存文件价格。

1. 基站存储容量

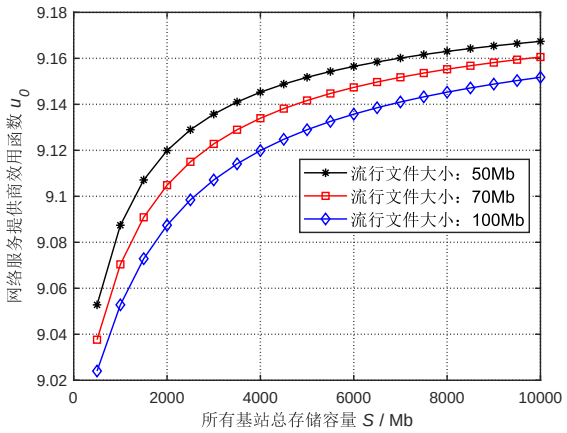

图 2: 网络服务提供商效用-不同基站存储容量

这里，我们分析不同的基站总存储容量 S 对于网络服务提供商效用函数 u_0 的影响。简单来说，总存储容量 S 的增加，一方面会增加缓存文件的数量，而网络服务提供商的收入 R_0 主要来源于该项，所以网络服务提供商的效用肯定会增加；但是另一方面也会导致维护成本 C_0 的逐渐升高。在图 2中，我们分别分析了流行文件大小等于 50Mb、70Mb 和 100Mb 时，网络服务提供商效用函数随着总存储容量 S 的变化曲线。三条曲线有着共同的变化趋势，在最开始，随着总存储容量 S 的增加，缓存文件数增加，收入 R_0 的增长快于维护成本 C_0，到了后期，维护成本 C_0 的增长速度也逐渐变快，因此出现了曲线逐渐平缓的趋势。

2. 缓存文件价格

由于网络服务提供商的收入 R_0 只和缓存文件价格相关，因此就缓存文件价格同网络服务提供商效用的关系进行分析。我们同样考虑流行文件大小等于 50Mb、70Mb 和 100Mb 的三种情况，并且严格按照不等式（28）中缓存文件价格的大小范围进行仿真。图 3中，我们可以看到每一条曲线都对应有一个最大值，该值的存在证明了网络服务提供商效用函数 u_0 的凹性。随着价格的上升，一开始缓存文件的数量变化幅度不大时，网络服务提供商的收入 R_0 会大幅度提高，但是一旦缓存文件价格过高，缓存文件的数量会逐渐降低影响着网络服务提供商的收入。流行文件大小为 50Mb 时，最大效用对应最佳缓存文件价格为 0.2；流行文件大小为 70Mb 和 100Mb 时，最佳缓存文件价格为 0.25。

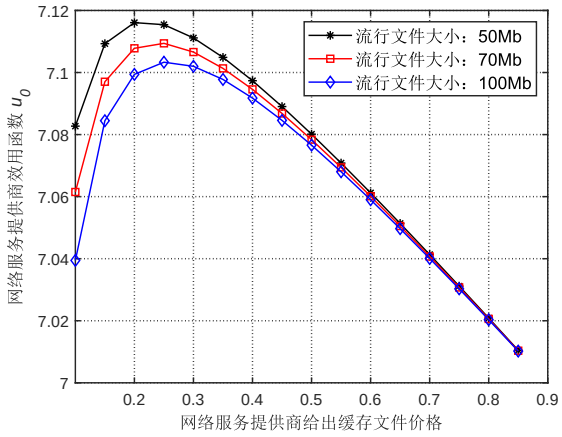

图 3: 网络服务提供商效用-不同缓存文件价格

5 结论

本文提出了一种基于斯塔克伯格博弈的基站间缓存分配技术。首先对该问题的系统模型进行分析构建，从整体网络架构到文件缓存模型，再到牵扯到网络服务提供商和流行内容提供商的利润模型。然后，根据斯塔克伯格博弈的概念将所研究问题建立为一个单领导者多跟随者博弈模型，其中对领导者和跟随者的优化问题分别进行分析及数学理论验证。之后针对多跟随着之间的非合作博弈提出了一种基站间缓存分配的异步算法。最后，通过相关仿真测试，验证了所提异步算法的快速收敛性，并且验证了之前所提出数学理论的正确性。

参考文献（References）

[1] Mach P, Becvar Z. Mobile edge computing: A survey on architecture and computation offloading[J]. IEEE Communications Surveys & Tutorials, 2017, 19(3): 1628-1656.

[2] Moura J, Hutchison D. Game theory for multi-access edge computing: Survey, use cases, and future trends[J]. IEEE Communications Surveys & Tutorials, 2018, 21(1): 260-288.

[3] Quijano N, Ocampo-Martinez C, Barreiro-Gomez J, et al. The role of population games and evolutionary dynamics in distributed control systems: The advantages of evolutionary game theory[J]. IEEE Control Systems Magazine, 2017, 37(1): 70-97.

[4] Messous M A, Senouci S M, Sedjelmaci H, et al. A game theory based efficient computation offloading in an UAV network[J]. IEEE Transactions on Vehicular Technology, 2019, 68(5): 4964-4974.

[5] Tang Q, Xie R, Huang T, et al. Jointly caching and computation resource allocation for mobile edge networks[J]. IET Networks, 2019, 8(5): 329-338.

[6] Pantisano F, Bennis M, Saad W, et al. In-network caching and content placement in cooperative small cell networks[C]//1st International Conference on 5G for Ubiquitous Connectivity. IEEE, 2014: 128-133.