

文章编号:1007-5321(2020)02-0029-11

DOI:10.13190/j.jbupt.2019-104

基于平均场博弈的超密集网络边缘缓存和删除分配研究

王孟哲¹, 滕颖蕾¹, 宋梅¹, 韩丹涛², 张勇¹

(1. 北京邮电大学 电子工程学院, 北京 100876; 2. 机械工业仪器仪表综合技术经济研究所, 北京 100055)

摘要: 超密集网络设备数目庞大导致缓存分配算法复杂度极高, 频繁地缓存和删除同样的内容导致的系统不稳定, 为此, 提出了基于平均场博弈(MFG)的分布式缓存分配算法和基于李雅普诺夫漂移加惩罚(DPP)方法的分布式删除分配算法. MFG方法使缓存分配算法的复杂度与基站数目无关. DPP方法将具有时间相关性的删除分配问题解耦成为每个时刻的问题, 并求解得到了兼顾系统稳定性和网络开销优化的删除分配策略. 仿真结果表明, MFG方法能够使网络最优控制策略快速收敛, 并且在超密集场景下得到明显低于基本缓存分配方法的网络开销; 李雅普诺夫 DPP方法能够实现兼顾网络开销优化的网络缓存和删除稳定性.

关键词: 超密集网络; 平均场; 平均场博弈; 李雅普诺夫理论; 漂移加惩罚

中图分类号: TN929.53

文献标志码: A

Mean-Field Game Based Edge Caching and Deleting Allocation in Ultra-Dense Networks

WANG Meng-zhe¹, TENG Ying-lei¹, SONG Mei¹, HAN Dan-tao², ZHANG Yong¹

(1. School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China;

2. Instrument Technology and Economy Institute, Beijing 100055, China)

Abstract: A distributed caching allocation algorithm based on mean-field game (MFG) and a distributed deleting allocation algorithm based on Lyapunov drift-plus-penalty (DPP) method were proposed to solve the problem of caching allocation algorithms' extremely high complexity caused by lots of devices in ultra-dense network and the problem of system instability caused by caching and deleting same content frequently. The caching allocation algorithm's complexity independent of the number of base stations is made by MFG. The time-correlated deleting allocation problem into problems each time slot is decoupled by DPP. Thereafter the deleting allocation policy for the tradeoff between system stability and minimizing network cost gets solved. Simulation shows that MFG can both make the network optimal control strategy converge quickly and save network cost obviously compared to baseline caching allocation method under ultra-dense scenario. Lyapunov DPP method can ensure network caching and deleting stability while minimizing network cost.

Key words: ultra-dense network; mean-field; mean-field game; Lyapunov theory; drift-plus-penalty

收稿日期: 2019-05-31

基金项目: 国家自然科学基金项目(61771072); 科技部重点研发基金项目(2018YFB120022)

作者简介: 王孟哲(1994—), 男, 硕士生.

通信作者: 滕颖蕾(1983—), 女, 副教授, E-mail: lilytengt@gmail.com.

边缘缓存网络中,当一个基站在制定缓存分配策略时,既要考虑存储受限的内容重叠,又要考虑容量受限的链路干扰问题. 在基站数目庞大的超密集网络中势必造成基站间信息的高度耦合. 一方面,“内容重叠”^[1]需要涉及全部其他基站的缓存分配策略;另一方面,在计算下行链路干扰时考虑全部其他基站的传输功率和信道增益. 此外,频繁地缓存和删除同一内容导致的系统不稳定也是一个容易被忽视的问题. 因此,兼顾系统稳定性的联合缓存和删除分配机制有待进一步研究.

近年来,平均场(MF, mean-field)理论在超密集网络的研究中得到了广泛的关注^[2-3]. 基于MF理论,可以通过统计平均效果来研究大量相互作用的个体的行为^[4]. MF理论可以将多个个体的问题转化为单一个体的问题,故可以在设备数目庞大的超密集网络中大幅降低网络优化的复杂度. Hamidouche等^[5]研究了基于平均场博弈(MFG, mean-field game)方法的超密集网络分布式缓存分配机制,虽然其运用MFG方法对内容重叠实现了低复杂度的计算,但在干扰的计算上没有做任何处理. Kim等^[6]采用MFG方法对内容重叠实现了低复杂度的计算,但在干扰的计算方面,采用的是随机几何方法,仅考虑了干扰的空间动态特性,忽视了干扰的时间动态特性,故不能很好地适应信道状态信息(CSI, channel state information)随时间的演化. 所以,对于MF理论在内容重叠计算和干扰计算上的应用的结合有待实现.

基于李雅普诺夫理论的漂移加惩罚(DPP, drift-plus-penalty)方法可以很好地解决边缘缓存网络的系统稳定性问题^[7-8]. DPP方法可以将一个优化一段时间的平均值的随机优化问题简化为多个优化每时刻瞬时值的确定性优化问题. Samarakoon等^[9]采用DPP方法依照队列状态信息进行用户调度,实现了系统的队列稳定性. 张琪^[10]针对与稳定性相关的缓存控制和接入控制变量分别构造了虚拟队列,并依照虚拟队列进行缓存控制和接入控制,最终实现了系统稳定性.

针对上述问题,笔者提出了基于MFG的分布式缓存分配策略和基于DPP方法的分布式删除分配策略. 每个基站通过独立地控制其缓存和删除分配策略,实现预定义的开销函数的长时间平均(LRA, long run average)值的最小化.

1 系统模型和问题构造

1.1 系统模型

考虑一个超密集网络,系统包含 K 个基站,集合为 $\mathcal{K} = \{1, 2, \dots, K\}$,系统中有 J 个待缓存文件,集合为 $\mathcal{J} = \{1, 2, \dots, J\}$. 文件 $j \in \mathcal{J}$ 的大小为 L_j ,假定所有基站对文件 j 的可用缓存空间均为 C_j ,向量 $\mathbf{P}_k(t) = [p_{k,1}(t), p_{k,2}(t), \dots, p_{k,J}(t)]$ 的分量 $p_{k,j}(t)$ 表示基站 $k \in \mathcal{K}$ 对文件 j 的缓存速率,服务中心对文件 j 分配的最大传输速率为 $B_j(t)$,故 $p_{k,j}(t) \in [0, B_j(t)]$; $\mathbf{E}_k(t) = [e_{k,1}(t), e_{k,2}(t), \dots, e_{k,J}(t)]$ 的分量 $e_{k,j}(t)$ 表示基站 k 对文件 j 的删除速率,受制于基站的硬件条件限制, $e_{k,j}(t) \in [0, E^{\max}]$; $Q_{k,j}(t) \in [0, C_j]$ 表示基站 k 对文件 j 的剩余缓存空间,其动态过程为

$$dQ_{k,j}(t) = [e_{k,j}(t) - p_{k,j}(t)]dt \quad (1)$$

假定用户只接入与之距离小于 R 的基站,在与之距离小于 R 的基站集合中,用户选择对其当前请求文件内容缓存最多的基站. 当没有基站缓存了用户当前的请求文件时,用户选择距离最近的基站接入. 令 $x_{k,j}(t)$ 为基站 k 附近文件 j 的流行度^[6],其动态变化服从均值为 μ_j 、变速参数为 r 的奥恩斯坦-乌伦贝克过程,有

$$dx_{k,j}(t) = r(\mu_j - x_{k,j}(t))dt + \eta dW_{k,j}(t) \quad (2)$$

其中: $\eta > 0$ 为常数, $W_{k,j}(t)$ 为维纳过程.

系统CSI^[5]的演化过程为

$$dh_k(t) = G(t, h_k(t))dt + \zeta dz_k(t) \quad (3)$$

其中: $G(t, h_k(t))$ 考虑了路径损耗和阴影, $\zeta dz_k(t)$ 包含衰落和信道不确定性. 假定所有基站以等功率 P 向用户传输信息,基站 k 到其服务用户的下行链路干扰为

$$I_k(t) = \sum_{k' \in \mathcal{K} \setminus \{k\}} P |h_{k'}(t)|^2 \quad (4)$$

则系统下行链路速率为

$$R_k(t) = W \lg \left(1 + \frac{P |h_k(t)|^2}{\sum_{k' \in \mathcal{K} \setminus \{k\}} P |h_{k'}(t)|^2 + \sigma^2} \right) \quad (5)$$

其中 σ^2 为高斯白噪声功率.

在边缘缓存网络中瞬时开销函数综合考虑了网络回程链路开销、基站的剩余存储空间、下行链路速率以及不同基站间的缓存内容重叠. 其中回程链路缓存开销^[5]为

$$\phi_{k,j}(p_{k,j}(t)) = -\lg\left(\frac{B_j(t) - p_{k,j}(t)}{B_j(t)}\right) \quad (6)$$

由于缓存的文件占据基站的存储空间,这会导致处理时延以及文件查询时延^[11],故定义与缓存大小成正比(不妨设比例系数为 γ)的文件存储开销:

$$\psi_{k,j}(Q_{k,j}(t)) = \gamma[C_j - Q_{k,j}(t)]/C_j \quad (7)$$

定义内容重叠函数:

$$I_{k,j}(\mathbf{P}_{-k,j}(t)) = \frac{1}{C_j L_j K} \sum_{k' \in \mathcal{K} \setminus \{k\}} p_{k',j}(t) \quad (8)$$

该函数用于描述一个文件在所有基站中的重复缓存程度. 全局瞬时开销函数为

$$J_{k,j}(t) = \frac{\phi_{k,j}(p_{k,j}(t)) [1 + I_{k,j}(\mathbf{P}_{-k,j}(t))]}{R_k(t) x_{k,j}(t)} + \psi_{k,j}(Q_{k,j}(t)) \quad (9)$$

1.2 问题构造

每个基站 k 的目标是在保证系统稳定性的条件下,通过对每个文件 j 控制缓存分配 $p_{k,j}(t)$ 和删除分配 $e_{k,j}(t)$,最小化对该文件的 LRA 开销. 优化问题可以写成

$$\mathcal{P}: \min_{(p_{k,j}, e_{k,j})} \lim_{t \rightarrow \infty} \frac{1}{t} \int_0^t J_{k,j}(\tau) d\tau \quad (10a)$$

$$\text{s. t 式(1) ~ 式(3)} \quad (10b)$$

$$p_{k,j}(t) \in [0, B_j(t)] \quad (10c)$$

$$e_{k,j}(t) \in [0, E^{\max}] \quad (10d)$$

$$\lim_{t \rightarrow \infty} \frac{1}{t} \int_0^t e_{k,j}(\tau) d\tau < \gamma_e \quad (10e)$$

其中:约束式(10e)表示系统的稳定性,即一个基站对于同一个文件不得频繁地缓存和删除; γ_e 表示 LRA 删除的上界. 虽然式(10e)本身和缓存速率无关,但只要删除不频繁,由约束式(1)可得缓存频率自然会随之降低.

考虑到缓存分配受制于服务中心到基站的回程容量和基站自身的开销最小化目标,其速度较慢,既需要通过每时每刻的积累才能实现对文件的缓存,也需要随着约束式(1)~式(3)随时调整. 而删除分配是基站内部的自我管理行为,且开销几乎为0,所以不受任何限制,速度非常快(即 E^{\max} 非常大,以至于 $e_{k,j}(t) \in [0, E^{\max}]$ 的限制可以忽略不计),并且删除需要相对于缓存有一定的迟滞,即缓存后一段时间再进行删除,以防删除过早导致的未来短时间内所需文件的丢失. 故需要在缓存和删除分配之间进行时间尺度分离,缓存分配是时刻变化的,为短时优化,而删除分配每隔时间 T 进行一次即可,为长时优化,每次制定的删除分配重新定义为 $\hat{e}_{k,j}(t)$,

$t = T, 2T, \dots$,它等效于 $\int_{t-T}^t e_{k,j}(\tau) d\tau$,其物理意义是 t 时刻删除文件 j 的大小,则取值范围为 $\hat{e}_{k,j}(t) \in [0, C_j - Q_{k,j}(t)]$.

2 短时优化问题:基于 MF 方法的缓存分配

由前面的重定义删除分配可知,只在 $t = T, 2T, \dots$ 时刻进行删除,故每次短时优化的时间段内没有删除分配,式(1)改写为式(11c).

短时优化的目标是基站对每个文件制定缓存策略. 短时优化问题为(不妨设短时优化的时间段为 $[0, T]$)

$$\mathcal{P}_s: \min_{p_{k,j}} \Gamma_{k,j}(0) \quad (11a)$$

$$\text{s. t 式(2), 式(3)} \quad (11b)$$

$$dQ_{k,j}(t) = -p_{k,j}(t) dt \quad (11c)$$

$$p_{k,j}(t) \in [0, B_j(t)] \quad (11d)$$

其中 $\Gamma_{k,j}(t)$ 表示 $[t, T]$ 时间段内开销的积分,有

$$\Gamma_{k,j}(t) = \int_t^T J_{k,j}(p_{k,j}(\tau), \mathbf{P}_{-k,j}(\tau)) d\tau \quad (12)$$

由于基站的状态随时间的动态演化由式(2)、式(3)和式(11c)所示,每个基站的缓存分配必须随之适应. 因此,在时间动态下最小化 LRA 开销的问题可以建模为动态随机微分博弈^[12](SDG, stochastic differential game)(一个文件一个博弈,且不同的文件相互独立). 博弈的“局中人”(player, 博弈的要素之一)为基站,基站的状态为 $\mathbf{S}_{k,j}(t) = [h_k(t), Q_{k,j}(t), x_{k,j}(t)]$, $\forall k \in \mathcal{K}$,对于文件 j 的 SDG 由 $(\mathcal{K}, \mathcal{S}_{k,j}, \mathcal{A}_{k,j}, \Gamma_{k,j})$ 表示,其中 $\mathcal{S}_{k,j}$ 是基站 k 的状态空间, $\mathcal{A}_{k,j}$ 是全部的缓存控制策略的集合 $\{p_{k,j}(t), 0 \leq t \leq T\}$. 根据文献[13],问题 \mathcal{P}_s 理论上可以通过逆向归纳法求解,即对于每个文件 j ,通过求解 K 个耦合的哈密顿-雅可比-贝尔曼(HJB, Hamilton-Jacobi-Bellman)方程式(13),能够事先(在0时刻)确定 LRA 开销的最小值.

$$0 = \frac{\partial \Gamma_{k,j}(h_k, Q_{k,j}, x_{k,j}, t)}{\partial t} +$$

$$\min_{p_{k,j}(t)} \left[J_{k,j}(p_{k,j}(t), \mathbf{P}_{-k,j}(t)) + G(t, h_k(t)) \frac{\partial \Gamma_{k,j}}{\partial h_k} - p_{k,j}(t) \frac{\partial \Gamma_{k,j}}{\partial Q_{k,j}} + r(\mu_j - x_{k,j}(t)) \frac{\partial \Gamma_{k,j}}{\partial x_{k,j}} \right]$$

$$\left[\frac{\zeta^2}{2} \frac{\partial^2 \Gamma_{k,j}}{\partial h_k^2} + \frac{\eta^2}{2} \frac{\partial^2 \Gamma_{k,j}}{\partial x_{k,j}^2} \right] \quad (13)$$

若式(13)对于全部基站有公共解,则问题 \mathcal{P}_s 存在纳什均衡^[14].

定义 1 对于一个给定全部基站的缓存控制策略组合 $\{p_{1,j}^*(t), p_{2,j}^*(t), \dots, p_{K,j}^*(t)\}$, 若对于任意基站 k , 任意可能的缓存控制策略组合 $\{p_{1,j}(t), p_{2,j}(t), \dots, p_{K,j}(t)\}$ (其中, 对于 $\forall k \in \mathcal{K}$, 都有 $p_{k,j}(t) \in \mathcal{A}_{k,j}$) 都满足:

$$\int_0^T J_{k,j}(p_{k,j}^*(\tau), \mathbf{P}_{-k,j}^*(\tau)) d\tau \leq \int_0^T J_{k,j}(p_{k,j}(\tau), \mathbf{P}_{-k,j}^*(\tau)) d\tau \quad (14)$$

那么 $\{p_{1,j}^*(t), p_{2,j}^*(t), \dots, p_{K,j}^*(t)\}$ 是纳什均衡.

由于定义时间动态的漂移函数式(13)中一阶导数的系数以及开销函数式(9)均具备平滑性, 式(13)对于全部基站有唯一公共解^[13]及其相应的唯一纳什均衡. 然而, 在 K 个 HJB 方程的交织系统的求解中, 每个基站都需要考虑全部其他基站的缓存分配和信道增益, 所以当基站数目 $K > 2$ 时, 该方法有着极大的计算复杂度. 然而, 当 K 足够大, 并且基站满足对于缓存控制策略的可交换性(同一时刻, 不同的基站在相同的自身状态下控制策略相同)时, 可以将该 SDG 转化为 MFG, 并通过迭代收敛至纳什均衡.

2.1 基于 MFG 的问题求解

MF 是所有“局中人”(基站)对状态(CSI、剩余缓存空间以及流行度)的分布密度, 这里由 $m_j(h, Q_j, x_j, t)$ 表示, 其定义为

$$m_j(h, Q_j, x_j, t) = \lim_{K \rightarrow \infty} \frac{1}{K} \sum_{k \in K} \delta\{h_k(t) = h, Q_{k,j}(t) = Q_j, x_{k,j}(t) = x_j\} \quad (15)$$

MF 由福克-普朗克-柯尔莫哥洛夫(FPK, Fokker-Planck-Kolmogorov)方程式(16)求得.

$$0 = \frac{\partial m_j}{\partial t} + G(t, h) \frac{\partial m_j}{\partial h} - p_j(h, Q_j, x_j, t) \frac{\partial m_j}{\partial Q_j} + r(\mu_j - x_j) \frac{\partial m_j}{\partial x_j} - \frac{\zeta^2}{2} \frac{\partial^2 m_j}{\partial h^2} - \frac{\eta^2}{2} \frac{\partial^2 m_j}{\partial x_{k,j}^2} \quad (16)$$

注意, 式(16)中的 $p_j(h, Q_j, x_j, t)$ 对应前面的可交换性的假设. 求得了 MF $m_j^*(h, Q_j, x_j, t)$, 干扰和内容重叠函数可由式(17)计算.

$$I_k(t) = \sum_{k' \in \mathcal{K} \setminus \{k\}} P|h_{k'}(t)|^2 \approx \sum_{k' \in K} P|h_{k'}(t)|^2 \approx KP \int h^2 m_j^* dh dQ_j dx_j = I(m_j^*) = I^*(t) \quad (17a)$$

$$I_{k,j}(\mathbf{P}_{-k,j}(t)) = \frac{1}{C_j L_j K} \sum_{k' \in \mathcal{K} \setminus \{k\}} p_{k',j}(t) \approx \frac{1}{C_j L_j K} \sum_{k' \in \mathcal{K}} p_{k',j}(t) \approx \frac{1}{C_j L_j} \int p_j(h, Q_j, x_j, t) m_j^* dh dQ_j dx_j = I_j^*(m_j^*) = I_j^*(t) \quad (17b)$$

此时基站已经不需要考虑其他基站的状态和缓存控制策略了, 只需知道 MF. 此时式(12)可以改写为

$$\Gamma_j(t) = \int_t^T J_j(p_j(h, Q_j, x_j, t), I^*(t), I_j^*(t)) d\tau \quad (18)$$

将式(18)代入式(13)得到 MF-HJB 方程:

$$0 = \frac{\partial \Gamma_j(h, Q_j, x_j, t)}{\partial t} + \min_{p_j(h, Q_j, x_j, t)} \left[J_j(p_j(h, Q_j, x_j, t), I^*(t), I_j^*(t)) + G(t, h) \frac{\partial \Gamma_j}{\partial h} - p_j(h, Q_j, x_j, t) \frac{\partial \Gamma_j}{\partial Q_j} + r(\mu_j - x_j) \frac{\partial \Gamma_j}{\partial x_j} + \frac{\zeta^2}{2} \frac{\partial^2 \Gamma_j}{\partial h^2} + \frac{\eta^2}{2} \frac{\partial^2 \Gamma_j}{\partial x_{k,j}^2} \right] \quad (19)$$

这是一个单独的 HJB 方程, 式(13)中的耦合已经完全不存在了. 通过逆向归纳法求解式(19)可以得到 LRA 开销的最优迹 $\Gamma_j^*(h, Q_j, x_j, t)$, 其对应的 MF $m_j^*(h, Q_j, x_j, t)$ 则是通过正向归纳法求解式(16)的结果, 这一对解 $[m_j^*(h, Q_j, x_j, t), \Gamma_j^*(h, Q_j, x_j, t)]$ 构成纳什均衡. 定义 $p_j^*(h, Q_j, x_j, t)$ 为由开销最优迹 $\Gamma_j^*(h, Q_j, x_j, t)$ 和 MF 分布 $m_j^*(h, Q_j, x_j, t)$ 生成的达到纳什均衡的最优缓存控制策略. 最优解 $p_j^*(h, Q_j, x_j, t)$ 由命题 1 给出.

命题 1 基站最优缓存控制策略为

$$p_j^*(h, Q_j, x_j, t) = B_j(t) - \frac{1 + I_j^*(t)}{R(h, I^*(t)) x_j} \frac{\partial \Gamma_j}{\partial Q_j} \quad (20)$$

其中 $I_j^*(t)$ 、 $I^*(t)$ 和 $\partial \Gamma_j / \partial Q_j$ 分别是式(17)和式(19)的唯一解. 其证明过程见附录.

考虑到求解 FPK 方程需要给定 LRA 开销的最优迹(HJB 方程的解)对应的最优缓存控制策略, 求解 HJB 方程需要给定 MF(FPK 方程的解), 所以需要通过迭代求解这对耦合的方程.

2.2 HJB-FPK 方程组的求解算法

采用不动点论证法^[15]求解耦合 HJB-FPK 方程

组的收敛解. 为了能够仿真求解偏微分方程, 需要采用 $\Delta t \rightarrow 0$ 的离散时间模拟连续时间, 积分也用求和的方式代替, 即 $\int_0^T f(t) dt \approx \lim_{\Delta t \rightarrow 0} \sum_{n=0}^{T/\Delta t-1} f(n\Delta t) \Delta t$. MF-HJB 方程需要在反向时间方向进行求解, 而 MF-FPK 方程需要在正向时间方向进行求解. 定义时间和状态的步长为 Δt 、 Δh 、 ΔQ_j 和 Δx_j . 为了保证不动点论证法的稳定性^[16], 步长需要满足式(21). 离散化之后, 对于左导数(反方向)、右导数(正方向)和二阶导数的模拟方法为式(22).

$$\begin{aligned} \frac{h_{\max} \Delta t}{(\Delta h)^2} &\leq \frac{1}{2} \\ \frac{Q_{j\max} \Delta t}{(\Delta Q_j)^2} &\leq \frac{1}{2} \\ \frac{x_{j\max} \Delta t}{(\Delta x_j)^2} &\leq \frac{1}{2} \end{aligned} \quad (21)$$

$$\begin{aligned} \frac{df(x)}{dx} &= \frac{f(x) - f(x - \Delta x)}{\Delta x} \\ \frac{df(x)}{dx} &= \frac{f(x + \Delta x) - f(x)}{\Delta x} \\ \frac{d^2f(x)}{dx^2} &= \frac{f(x + \Delta x) - 2f(x) - f(x - \Delta x)}{(\Delta x)^2} \end{aligned} \quad (22)$$

对于算法实现, 以下步骤为先决条件:

- 1) 通过式(22)离散化 MF-HJB 方程式(19)和 MF-FPK 方程式(16);
- 2) 通过式(22)离散化方程式(2)、式(3)和式(11c);
- 3) 为 Δt 、 Δh 、 ΔQ_j 和 Δx_j 设定符合条件式(21)的值.

完成了上述准备工作, 就可以通过算法1求解 HJB-FPK 方程组, 并且基于算法1的解, 每个基站可以通过算法2求解自己的最优控制策略和相应的最低开销函数.

算法1 不动点论证法求解 HJB-FPK 方程组

输入: 短时优化时长 T , 收敛精度 ε , $p_j^0(h, Q_j, x_j, t)$, 0时刻 MF $m_j^i(h, Q_j, x_j, 0)$.

步骤1 求解式(16), 解由 $m_j^0(h, Q_j, x_j, t)$ 表示;

步骤2 $i = 1, d = 1$;

步骤3 将 $m_j^{i-1}(h, Q_j, x_j, t)$ 代入式(17)求得干扰和内容重叠函数, 并代入式(19), 在反向时间方向上求解 MF-HJB 方程, 解由 $\Gamma_j^i(h, Q_j, x_j, t)$ 表示;

步骤4 根据式(20)求解最优缓存控制策略,

解由 $p_j^i(h, Q_j, x_j, t)$ 表示, 其中

$$\frac{\partial \Gamma_j^i}{\partial Q_j} \approx \frac{\Gamma_j^i(h, Q_j + \Delta Q_j, x_j, t) - \Gamma_j^i(h, Q_j, x_j, t)}{\Delta Q_j}$$

步骤5 将 $p_j^i(h, Q_j, x_j, t)$ 代入式(16), 在正向时间方向上求解 MF-FPK 方程, 解由 $m_j^i(h, Q_j, x_j, t)$ 表示;

步骤6 $d = |m_j^i(h, Q_j, x_j, t) - m_j^{i-1}(h, Q_j, x_j, t)|$, $i = i + 1$;

步骤7 检查 d , 如果不满足条件 $d < \varepsilon$, 则返回步骤3, 否则终止.

输出: 纳什均衡 $m_j(h, Q_j, x_j, t)$, $\Gamma_j(h, Q_j, x_j, t)$.

算法2 一个基站的分布式缓存控制策略

输入: $m_j(h, Q_j, x_j, t)$, $Q_{k,j}(0)$, $x_{k,j}(0)$, $h_k(0)$, 随机过程 $W_{k,j}, z_k$.

步骤1 将 $W_{k,j}, z_k$ 代入式(2)和式(3)求得 $x_{k,j}(t)$, $h_k(t)$;

步骤2 $\Gamma_{k,j}(T) = 0, N = T/\Delta t, n = 0$;

步骤3 将 $p_{k,j}(n\Delta t)$ 代入 MF-HJB 方程式(19) (先代入, 暂不求解), 代入式(11c)导出 $Q_{k,j}((n+1)\Delta t)$;

步骤4 根据式(17)计算干扰 $I((n+1)\Delta t)$ 和内容重叠函数 $\Gamma_j((n+1)\Delta t)$, 并代入 MF-HJB 方程式(19)求得 $\Gamma_{k,j}((n+1)\Delta t)$;

步骤5 将步骤4的解代入式(20)求得 $p_{k,j}((n+1)\Delta t)$, 其中 $\frac{\partial \Gamma_{k,j}}{\partial Q_{k,j}} \approx \frac{\Gamma_{k,j}((n+1)\Delta t) - \Gamma_{k,j}(n\Delta t)}{Q_{k,j}((n+1)\Delta t) - Q_{k,j}(n\Delta t)}$;

步骤6 $n = n + 1$;

步骤7 检查 n , 如果不满足条件 $n = N$, 则返回步骤3, 否则终止.

输出: $\Gamma_{k,j}(t)$, $p_{k,j}(t)$.

3 长时优化问题: 基于李雅普诺夫理论的删除分配

长时优化的目标是基站对文件制定删除策略. 考虑到长时优化问题每隔时间 T 进行一次, 时间已经离散化(不妨设 $T = 1$, 将时间归一化), 以及第1节对删除策略的重定义, 并且开销中只有存储开销和删除策略有关, 则长时优化的子问题为

$$\mathcal{P}_s: \min_{p_{k,j}} \Gamma_{k,j}(0) \quad (23a)$$

$$\text{s. t. } Q_{k,j}(t+1) = Q_{k,j}(t) + \hat{e}_{k,j}(t) - \hat{p}_{k,j}(t) \quad (23b)$$

$$\hat{e}_{k,j}(t) \in [0, C_j - Q_{k,j}(t)] \quad (23c)$$

$$\lim_{t \rightarrow \infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \hat{e}_{k,j}(\tau) < \gamma_e \quad (23d)$$

其中 $\hat{p}_{k,j}(t)$ 等效为原问题中的 $\int_t^{t+T} L_j p_{k,j}(\tau) d\tau$. 显然, $\hat{p}_{k,j}(t)$ 是下一段短时优化的解, t 时刻无法确定, 这个问题会在后面得到处理.

3.1 子问题转化

由李雅普诺夫理论, 构建虚拟队列:

$$\alpha_{k,j}(t+1) = \max[0, \alpha_{k,j}(t) + \hat{e}_{k,j}(t) - \gamma_e] \quad (24)$$

李雅普诺夫函数为 $L(t) = \frac{1}{2} \alpha_{k,j}^2(t)$, 则

$$\Delta L(t) = L(t+1) - L(t) =$$

$$\frac{1}{2} [\alpha_{k,j}^2(t+1) - \alpha_{k,j}^2(t)] \leq$$

$$\frac{1}{2} [\alpha_{k,j}(t) + \hat{e}_{k,j}(t) - \gamma_e]^2 - \alpha_{k,j}^2(t)] =$$

$$\frac{1}{2} [\alpha_{k,j}^2(t) + [\hat{e}_{k,j}(t) - \gamma_e]^2 +$$

$$2\alpha_{k,j}(t) [\hat{e}_{k,j}(t) - \gamma_e] - \alpha_{k,j}^2(t)] =$$

$$\frac{1}{2} [\hat{e}_{k,j}(t) - \gamma_e]^2 + \alpha_{k,j}(t) [\hat{e}_{k,j}(t) - \gamma_e] \leq$$

$$U + \alpha_{k,j}(t) [\hat{e}_{k,j}(t) - \gamma_e]$$

即

$$\Delta L(t) \leq U + \alpha_{k,j}(t) [\hat{e}_{k,j}(t) - \gamma_e] \quad (25)$$

其中 U 是 $\frac{1}{2} [\hat{e}_{k,j}(t) - \gamma_e]^2$ 的一致上界. 考虑到 t 时刻的状态已经确定, 惩罚函数的最小化需要针对删除控制变量 $\hat{e}_{k,j}(t)$, 所以惩罚函数为

$$p(t) = \psi_{k,j}(Q_{k,j}(t+1)) =$$

$$\psi_{k,j}(Q_{k,j}(t) + \hat{e}_{k,j}(t) - \hat{p}_{k,j}(t)) \quad (26)$$

由于 $\hat{p}_{k,j}(t)$ 是下一段短时优化的解, t 时刻无法确定, 但考虑到对于 $\hat{e}_{k,j}(t)$ 的取值来说, 最小化 $\psi_{k,j}(Q_{k,j}(t) + \hat{e}_{k,j}(t) - \hat{p}_{k,j}(t))$ 等同于最小化 $\psi_{k,j}(Q_{k,j}(t) + \hat{e}_{k,j}(t))$, 故惩罚函数可以改写为

$$p(t) = \psi_{k,j}(Q_{k,j}(t) + \hat{e}_{k,j}(t)) \quad (27)$$

李雅普诺夫 DPP 函数为

$$\Delta L(t) + Vp(t) \leq U + \alpha_{k,j}(t) [\hat{e}_{k,j}(t) - \gamma_e] +$$

$$V\psi_{k,j}(Q_{k,j}(t) + \hat{e}_{k,j}(t)) \quad (28)$$

其中 $V \geq 0$ 是系统稳定性和网络性能优化 (开销最小化) 之间的权衡系数. 注意到式 (28) 中有些项和控制变量 $\hat{e}_{k,j}(t)$ 无关, 问题式 (23) 转化为

$$\mathcal{P}_{11} : \min_{(\hat{e}_{k,j})} \alpha_{k,j}(t) \hat{e}_{k,j}(t) + V\psi_{k,j}(Q_{k,j}(t) + \hat{e}_{k,j}(t)) \quad (29a)$$

$$\text{s. t. } \hat{e}_{k,j}(t) \in [0, C_j - Q_{k,j}(t)] \quad (29b)$$

3.2 问题求解

对式 (29) 中的目标函数的删除控制变量求导得

$$\frac{\partial}{\partial \hat{e}_{k,j}(t)} \left[\alpha_{k,j}(t) \hat{e}_{k,j}(t) + V\psi_{k,j}(Q_{k,j}(t) + \hat{e}_{k,j}(t)) \right] = \alpha_{k,j}(t) - V \frac{\gamma}{C_j} \quad (30)$$

导数为常数, 故 $\alpha_{k,j}(t) \hat{e}_{k,j}(t) + V\psi_{k,j}(Q_{k,j}(t) + \hat{e}_{k,j}(t))$ 是 $\hat{e}_{k,j}(t)$ 的单调函数, 所以最终的最优删除策略为

$$\hat{e}_{k,j}(t) = \begin{cases} 0, & \alpha_{k,j}(t) - V \frac{\gamma}{C_j} \geq 0 \\ C_j - Q_{k,j}(t), & \text{其他} \end{cases} \quad (31)$$

4 仿真分析

该部分通过 Matlab 仿真评估提出的方法的性能. 假定基站的位置分布服从齐次泊松点过程, 基站的初始状态分布为正态分布. 在相邻两次删除分配之间, 基站进行 25 次缓存分配. 仿真参数如表 1 所示.

表 1 仿真参数

| 仿真参数 | 数值 |
|--|---------|
| 网络密度/km ⁻² | 0.45 |
| 基站容量 C _j /GB | 1 |
| 回程链路容量 B _j /(GB·s ⁻¹) | 1 |
| 文件大小 L _j /GB | 1 |
| 网络规模/km | 20 × 20 |
| 信道损失指数 | 2 |
| 信道带宽 W/MHz | 480 |
| 传输功率/dBm | 23 |
| 噪声功率/dBm | -70 |

4.1 MFG 方法的计算复杂度

图 1 展示了在不同的基站对于剩余存储空间的条件, MFG 方法的计算复杂度 (迭代次数) 与基站密度的关系. 无论初始分布如何, 迭代次数都大体上随基站密度单调递减, 并最终收敛至 3. 这表明, 网络越密集, MFG 方法的计算复杂度越低; 迭代次数的最终收敛表明了当网络足够密集时, MFG 方法的复杂度已经和网络密度无关了, 迭代次数仅仅为 3 也充分证明了 MFG 方法能够在超密集网络中实现非常低的复杂度. 所以, 上述分析充分说明了 MFG 方法对于降低超密集网络缓存分配方法的复杂度有着巨大的意义.

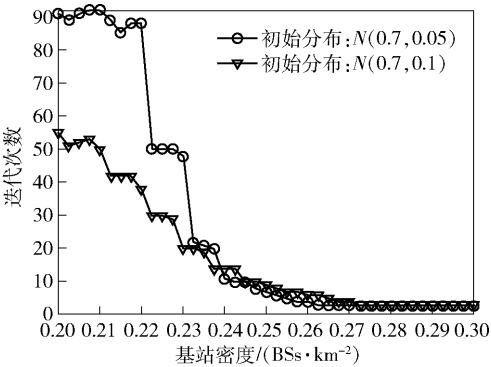


图 1 MF 方法迭代次数随基站密度的变化

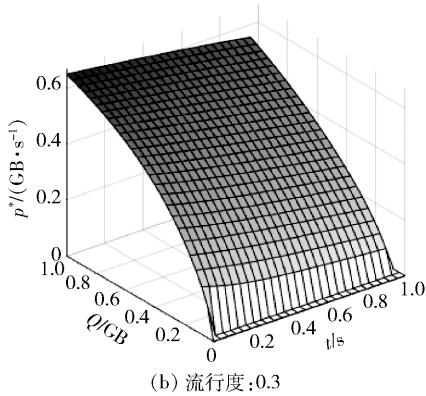
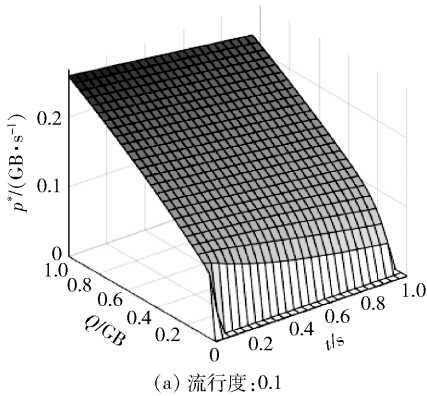


图 2 最优缓存策略

另一方面已有的缓存已经较多,所以对于下载的需求较低,此时缓存分配的重点是减少下载开销.此外,对于同一流行度、同一剩余存储空间状态,下载策略随时间单调递减.这是因为 MFG 方法的优化目标是网络开销在 $[0, T]$ 的积分,所以随着时间的推移,剩余的时间越来越少,缓存文件的意义也随之减小.上述的一切充分表明 MFG 方法在降低超密集网络缓存分配方法的复杂度的同时,既能够很

好地保证基站缓存策略适应其自身状态,也能够很好地兼顾网络开销的最小化.

4.3 MFG 方法的最优缓存策略

图 3 给出了 MF 分布随时间的变化.显然,当文件流行度较高时, MF 分布的变化也越大.这再一次印证了 MFG 方法的下载策略能很好地适应文件流行度, MFG 方法能够在降低超密集网络缓存分配方法的复杂度的同时很好地保证基站缓存策略适应其

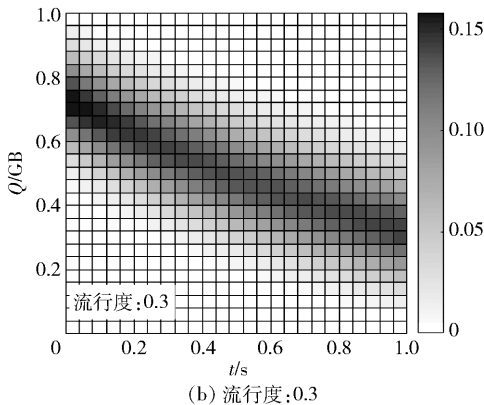
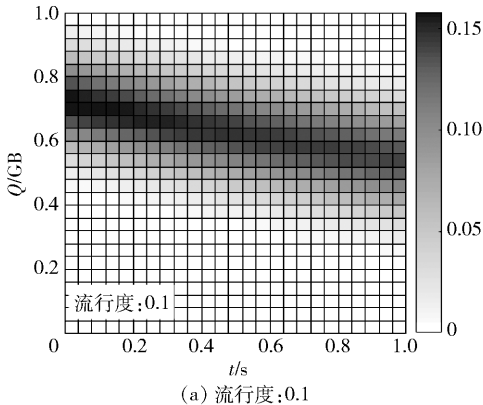


图 3 MF 分布随时间的演化

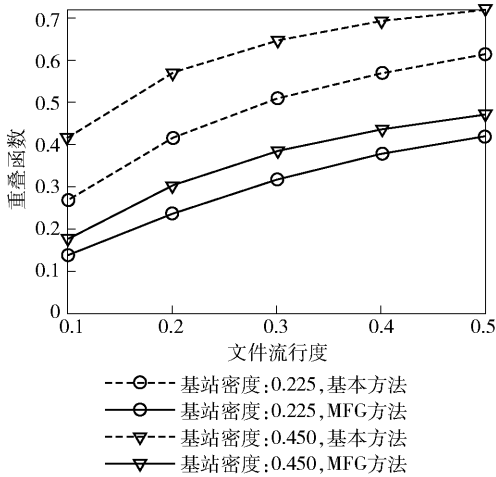
自身状态.

4.4 短时优化的性能比较

本节将 MFG 方法与基本缓存方法^[6]进行比较. 基本缓存方法脱离了对内容重叠的考虑, 在每个时刻基于当前的回程链路和存储状态, 制定与瞬时文件流行度成比例的下载策略, 如式(32)所示.

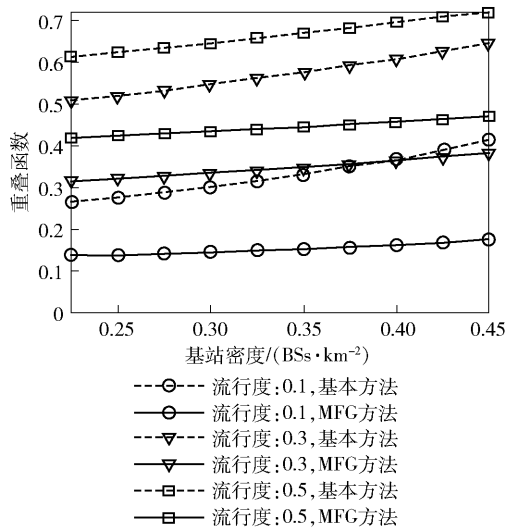
$$p_j^*(t) = B_j(t) - \frac{1}{1 + R_k(t)x_{k,j}(t)} \quad (32)$$

图 4(a)所示为 MFG 方法和基本方法的内容重叠函数与文件流行度的关系. 对于较低的文件流行度(0.1), MFG 方法能够比基本方法在 0.225 的网络密度下降低 48.0% 的内容重叠, 在 0.45 的



(a) 重叠函数与文件流行度的关系

网络密度下降低 57.8%. 然而, 随着文件流行度的增加, 相比于基本方法, MFG 方法带来的内容重叠下降率减小. 当文件流行度为 0.5 时, 在 0.225 的网络密度下降低 31.5%, 在 0.45 的网络密度下降低 34.8%. 这是因为当文件流行度较高时, 对于下载的需求也较高, 所以此时允许有较高的内容重叠, 即此时对于内容重叠降低的需求较低, 故此时 MFG 方法对于内容重叠的精确计算的优势相对不明显. 然而, 随着文件流行度的降低, 降低内容重叠以减少网络开销愈发成为缓存分配的重点, 所以 MFG 方法对于内容重叠的精确计算的优势愈发明显.



(b) 重叠函数与基站密度的关系

图 4 内容重叠函数比较

图 4(b)所示为内容重叠函数与网络密度的关系. 对于较低的网络密度(0.225), MFG 方法能够比基本方法在 0.1 的文件流行度下降低 48.0% 的内容重叠, 在 0.3 的文件流行度下降低 37.4%, 在 0.5 的文件流行度下降低 31.5%. 随着网络密度的增加, 相比于基本方法, MFG 方法带来的内容重叠下降率减小. 当网络密度为 0.45 时, 在 0.1 的文件流行度下降低 57.8%, 在 0.3 的文件流行度下降低 40.6%, 在 0.5 的文件流行度下降低 34.8%. 所以, 网络越密集, MFG 方法的优势越明显, 这表明了 MF 理论对于超密集网络的意义.

4.5 李雅普诺夫稳定性

图 5 在不同的 LRA 删除策略上界 γ_e 的条件下给出了不同文件流行度对应的 LRA 删除随时间的变化.

对于任意的文件流行度和 γ_e , LRA 删除都最终位于 γ_e 以下. 而在 LRA 删除策略随时间的变化趋势上, 有的是折线, 有的则是平滑的曲线并且趋近于直线. 在平滑的曲线上, 每一个点都是位于 γ_e 以下的, 即平滑的曲线对应的 LRA 删除在任意时刻都小于 γ_e , 而折线则不满足这一条件. 其原因如下: 采用李雅普诺夫 DPP 方法是为了保证系统的缓存-删除稳定性, 即约束条件式(10e). 若对于给定的删除策略上界 γ_e 来说, 当文件流行度较低时, 由于缓存的内容小于 γ_e , 此时即使不采用李雅普诺夫 DPP 方法, 也能够满足系统稳定性的约束条件(LRA 删除策略在 γ_e 以下), 故此时 DPP 方法不起作用. 例如, 当流行度为 0.3 或 0.5 时, 删除策略本身就低于 0.7, 故对于 $\gamma_e = 0.7$, DPP 方法不起作用. 若对于给定的删除策略上界 γ_e 来说, 文件流行度较高, 那么

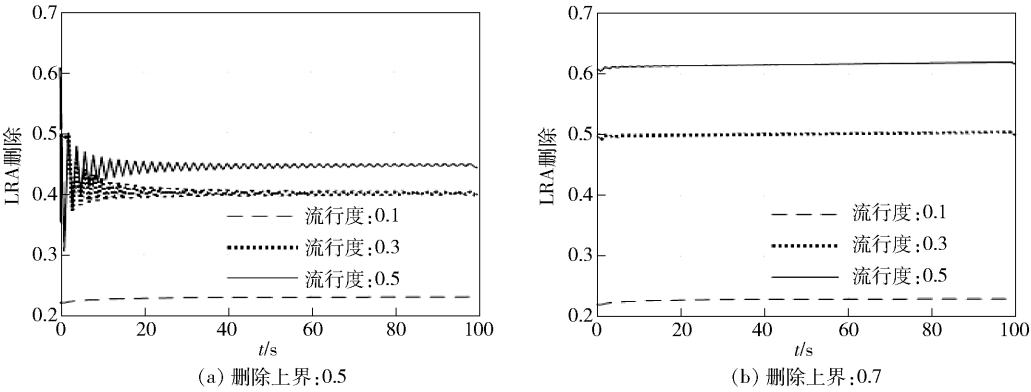


图 5 LRA 删除随时间的演化

由于缓存的内容大于 γ_e , 此时若不采用李雅普诺夫 DPP 方法, 则无法满足系统稳定性的约束条件. 例如, 当流行度为 0.3 或 0.5 时, 删除策略本身大于 0.5, 故当 γ_e 为 0.5 时, DPP 方法起作用.

4.6 总方法 LRA 开销比较

本节将提出的方法整体与一个基本方法进行比较. 基本方法中缓存分配方法已经在 4.4 节中提到, 删除分配方法不考虑缓存-删除稳定性, 仅仅制定最小化网络开销的删除策略.

图 6 给出了 LRA 回程链路开销和存储开销随

时间的变化. 需要注意的是, 流行度为 0.1 时, 提出的方法的 3 条线完全重合. 由 4.5 节可知, 这是因为此时 DPP 方法不起作用, 故开销与 γ_e 无关. 显然, 对于两种流行度, 提出的方法的 LRA 回程链路开销均完胜基本方法. 结合 4.5 节中 DPP 方法何时起作用的结论可知, 当 DPP 方法不起作用时, 回程链路开销几乎不随时间变化, 此时相比于基本方法的 LRA 回程链路开销的降低完全是 MFG 方法的功劳; 当 DPP 方法起作用时, 提出的方法的 LRA 回程链路开销先随时间递减, 随后几乎不随时间变化, 此

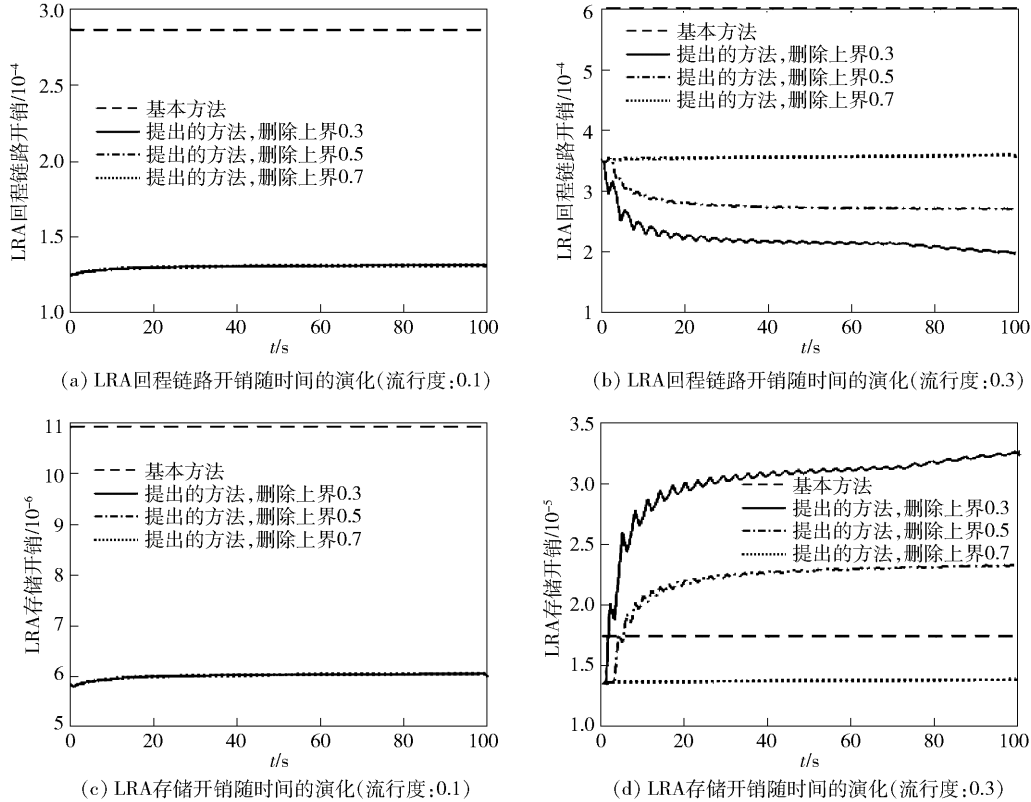


图 6 LRA 开销随时间的演化

时相比于基本方法的 LRA 回程链路开销的降低是 MFG 方法和 DPP 方法共同作用的结果。并且,对于给定的流行度,DPP 方法起作用时的 LRA 回程链路开销小于不起作用时的开销;另外,对于 DPP 方法起作用的场景下, γ_e 的值越低,LRA 回程链路开销越小。故采用 DPP 方法不仅可以保证文件的缓存-删除稳定性,也可以通过减少反复地缓存和删除来降低回程链路开销。

对于 LRA 存储开销,结合 4.5 节中 DPP 方法何时起作用的结论可知,当 DPP 方法不起作用时,提出的方法的 LRA 存储开销完胜基本方法,这完全是 MFG 方法的功劳;而当 DPP 方法起作用时,提出的方法的 LRA 存储开销先随时间递增,并很快地高于基本方法,随后几乎不随时间变化,故此时是 DPP 方法对存储开销的增加盖过了 MFG 方法对存储开销的降低。因为 DPP 方法减少了对文件频繁的缓存和删除,故必然会增加对文件的存储,从而增加存储开销,其价值在于在删除分配时对流行动度较高的文件进行一定程度的强制保留,且对于相同的流行度, γ_e 的值越高,DPP 方法起的作用越大,强制保留越积极,存储开销越大。结合前面的 DPP 方法对缓存-删除稳定性和回程链路开销降低的作用可知,DPP 方法是以牺牲存储开销为代价换取缓存-删除稳定性和回程链路开销的策略。

5 结束语

针对超密集网络中基于 MFG 和李雅普诺夫 DPP 方法的缓存和删除分配机制进行了研究。通过考虑 CSI、基站存储状态信息和文件流行度状态信息等因素,采用动态 SDG 的模型对优化问题进行建模,并将这个具有严重耦合的博弈进行解耦,将其转化为 MFG,通过分析研究得到超密集场景下最小化基站开销的缓存分配机制。在长时优化的文件删除分配策略上,根据李雅普诺夫理论将子问题转化为李雅普诺夫随机优化问题,并采用李雅普诺夫 DPP 方法实现了开销和系统稳定性之间的权衡。仿真结果表明,MFG 方法在实现超密集网络缓存分配策略快速收敛的同时,既能够保证缓存分配策略很好地适应文件流行度和剩余存储空间状态,也能够很好地兼顾网络开销的最小化;网络密度越大,MFG 方法越有意义;采用 DPP 方法能够在兼顾节约网络开销的同时保证系统的稳定性,且文件流行度越高,DPP 方法的作用越明显。

参考文献:

- [1] Chun B G, Chaudhuri K, Wee H, et al. Selfish caching in distributed systems: a game-theoretic analysis [M]. New York: Association for Computing Machinery, 2004: 21-30.
- [2] Yang C, Li J, Semasinghe P, et al. Distributed interference and energy-aware power control for ultra-dense D2D networks: a mean field game [J]. IEEE Transactions on Wireless Communications, 2017, 16(2): 1205-1217.
- [3] Al-Zahrani A, Yu F R, Huang M. A joint cross-layer and co-layer interference management scheme in hyper-dense heterogeneous networks using mean-field game theory [J]. IEEE Transactions on Vehicular Technology, 2016, 65(3): 1522-1535.
- [4] Baillieul J, Samad T. Encyclopedia of systems and control [M]. London: Springer London, 2014: 1-6.
- [5] Hamidouche K, Saad W, Debbah M, et al. Mean-field games for distributed caching in ultra-dense small cell networks [C] // American Control Conference. Boston: [s. n.], 2016: 4699-4704.
- [6] Kim H, Park J, Bennis M, et al. Ultra-dense edge caching under spatio-temporal demand and network dynamics [C] // IEEE International Conference on Communications (ICC). Paris: [s. n.], 2017: 1-7.
- [7] Neely M J. Stochastic network optimization with application to communication and queueing systems [J]. Synthesis Lectures on Communication Networks, 2010, 3(1): 1-211.
- [8] Huang L, Neely M J. Utility optimal scheduling in processing networks [J]. Performance Evaluation, 2011, 68(11): 1002-1021.
- [9] Samarakoon S, Bennis M, Saad W, et al. Ultra dense small cell networks: turning density into energy efficiency [J]. IEEE Journal on Selected Areas in Communications, 2016, 34(5): 1267-1280.
- [10] 张琪. 联合缓存与无线传输的理论性能分析及优化策略研究 [D]. 北京: 北京邮电大学, 2018.
- [11] Cao Y, Tao M, Xu F, et al. Fundamental storage-latency tradeoff in cache-aided MIMO interference networks [J]. IEEE Transactions on Wireless Communications, 2017, 16(8): 5061-5076.
- [12] Gueant O, Lasry J, Lions P. Mean field games and applications [M]. Berlin, Germany: Springer Berlin Heidelberg, 2011: 205-266.
- [13] Bernt O. Stochastic differential equations [J]. IEEE Transactions on Automatic Control, 2006, 51(10): 1731-1732.

- [14] Debbah M, Couillet R, Perlaza S M, et al. Electrical vehicles in the smart grid: a mean field game analysis [J]. IEEE Journal on Selected Areas in Communications, 2012, 30(6): 1086-1096.
- [15] Aziz M, Caines P E. A mean field game computational methodology for decentralized cellular network optimization[J]. IEEE Transactions on Control Systems Technology, 2017, 25(2): 563-576.
- [16] Li J, Chen Y T, Chen G, et al. Computational partial differential equations using Matlab (Chapman & Hall/CRC Applied Mathematics & Nonlinear Science) [M]. New York: Taylor & Francis, 2011.

附录: 最优缓存策略的证明过程

证明 MF-HJB 方程式(19)的最小值项显然有其对应的最小值点:

$$p_j^*(h, Q_j, x_j, t) = \arg \min_{p_j(h, Q_j, x_j, t)} \left[J_j(p_j(h, Q_j, x_j, t), I_j^*(t), I_j^{r*}(t)) + G(t, h) \frac{\partial \Gamma_j}{\partial h} - p_j(h, Q_j, x_j, t) \frac{\partial \Gamma_j}{\partial Q_j} + r(\mu_j - x_j) \frac{\partial \Gamma_j}{\partial x_j} + \frac{\xi^2}{2} \frac{\partial^2 \Gamma_j}{\partial h^2} + \frac{\eta^2}{2} \frac{\partial^2 \Gamma_j}{\partial x_j^2} \right]$$

中括号里面的表达式对 p_j 求偏导数得

$$\frac{\partial}{\partial p_j} \left[J_j(p_j(h, Q_j, x_j, t), I_j^*(t), I_j^{r*}(t)) + \right.$$

$$\begin{aligned} & G(t, h) \frac{\partial \Gamma_j}{\partial h} - p_j(h, Q_j, x_j, t) \frac{\partial \Gamma_j}{\partial Q_j} + \\ & r(\mu_j - x_j) \frac{\partial \Gamma_j}{\partial x_j} + \frac{\xi^2}{2} \frac{\partial^2 \Gamma_j}{\partial h^2} + \frac{\eta^2}{2} \frac{\partial^2 \Gamma_j}{\partial x_j^2} \Big] = \\ & \frac{\partial}{\partial p_j} \left[J_j(p_j(h, Q_j, x_j, t), I_j^*(t), I_j^{r*}(t)) - \right. \\ & \quad \left. p_j(h, Q_j, x_j, t) \frac{\partial \Gamma_j}{\partial Q_j} \right] = \\ & \frac{\partial J_j(p_j(h, Q_j, x_j, t), I_j^*(t), I_j^{r*}(t))}{\partial p_j} - \frac{\partial \Gamma_j(t)}{\partial Q_j} = \\ & \frac{\partial}{\partial p_j} \left[\frac{\phi_j(p_j(h, Q_j, x_j, t)) [1 + I_j^{r*}(t)]}{R(h, I_j^*(t)) x_j} + \right. \\ & \quad \left. \psi_j(Q_j) \right] - \frac{\partial \Gamma_j(t)}{\partial Q_j} = \\ & \frac{1 + I_j^{r*}(t)}{R(h, I_j^*(t)) x_j} \frac{\partial \phi_j(p_j(h, Q_j, x_j, t))}{\partial p_j} - \\ & \frac{\partial \Gamma_j(t)}{\partial Q_j} = \frac{1 + I_j^{r*}(t)}{R(h, I_j^*(t)) x_j} \frac{1}{B_j(t) - p_j(h, Q_j, x_j, t)} - \\ & \quad \frac{\partial \Gamma_j(t)}{\partial Q_j} = 0 \\ & p_j^*(h, Q_j, x_j, t) = B_j(t) - \frac{1 + I_j^{r*}(t)}{R(h, I_j^*(t)) x_j} \frac{\partial \Gamma_j}{\partial Q_j} \end{aligned}$$

证毕.