基于强化学习的用户关系组的缓存分配优化

作者姓名 轩夺

指导教师姓名、职称 陈健 教授

申请学位类别 工学硕士

学校代码 10701　　学　号 17011210011
分 类 号 TN92　　密　级 公开

西安电子科技大学

硕士学位论文

基于强化学习的用户关系组的缓存分配优化

作者姓名：轩夺

一级学科：信息与通信工程

二级学科（研究方向）：通信与信息系统

学位类别：工学硕士

指导教师姓名、职称：陈健　教授

学　　院：通信工程学院

提交日期：2020 年 4 月

Cache Allocation Optimization of User Relationship Group Based on Reinforcement Learning

A thesis submitted to

XIDIAN UNIVERSITY

in partial fulfillment of the requirements

for the degree of Master

in Information and Communications Engineering

By

Xuan Duo

Supervisor: Chen Jian Title: Professor

April 2020

西安电子科技大学
学位论文独创性（或创新性）声明

秉承学校严谨的学风和优良的科学道德，本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果；也不包含为获得西安电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同事对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文若有不实之处，本人承担一切法律责任。

本人签名：_____　　日　　期：_____

西安电子科技大学
关于论文使用授权的说明

本人完全了解西安电子科技大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权属于西安电子科技大学。学校有权保留送交论文的复印件，允许查阅、借阅论文；学校可以公布论文的全部或部分内容，允许采用影印、缩印或其它复制手段保存论文。同时本人保证，结合学位论文研究成果完成的论文、发明专利等成果，署名单位为西安电子科技大学。

保密的学位论文在___年解密后适用本授权书。

本人签名：_____　　导师签名：_____

日　　期：_____　　日　　期：_____

摘要

随着信息技术和移动设备的发展，特别是 4G 的普及以及 5G 的商用，视频等网络资源呈现爆发式增长。大规模的网络流量存在于网络中尤其在高峰期会造成网络回程拥挤，使得用户的请求延迟增大，降低量用户的上网体验。如何解决网络拥挤的问题是当下网络结构优化的重点。

为了缓解网络拥塞和降低请求延迟，D2D 缓存技术成为了一个现代通信网络重要的组成部分。人们总是在网上浏览他们感兴趣的内容。D2D 缓存技术可以缓存人们比较感兴趣的内容，使得人们在请求内容时可以直接通过 D2D 通信得到而不需要向基站请求。由于不同的人有不同的兴趣点，并且人们的兴趣点会随着时间和社会热点的出现发生变化，缓存设备的选取以及如何有效决定缓存内容成为了亟待解决的问题。现阶段的研究主要根据用户的设备能力和历史通信记录来决定缓存设备的选择。实际上缓存节点周围用户的请求分布是在不断变化的而且历史记录的数据量不是足够的，这使得缓存节点的选择和缓存的内容并不是固定不变的。

本文提出了一种基于虚拟时延的最优缓存节点选择算法，该算法以多臂赌博机模型为基础，根据用户的兴趣差异获得最优的缓存决策。每个候选用户均有机会成为一个缓存节点，该算法选择系统延迟最小的候选用户作为缓存节点。但是实际情况下，在一定的区域内，会同时存在多个缓存节点。在缓存节点空间有限的条件下，如何提高一定区域内请求者的请求缓存命中率是一个重要问题。现阶段的研究主要集中在多缓存节点之间进行协作的方式。由于每个缓存节点的缓存能力和周边用户的请求分布不同，导致每个缓存节点的服务能力有所差异，所以有必要对各个缓存节点的服务能力进行有效区分，进而充分利用各个缓存节点的服务能力，使得整体收益达到最大。因此本文提出了一种主从节点协作缓存算法。该算法以缓存节点选择算法为基础，充分利用服务能力较强的缓存节点之间的协作，调整各个缓存节点的缓存决策，使得整体请求延迟达最小。另外，考虑到实际情况下用户多具有移动性，本文在移动场景下测试了提出的基于虚拟时延的最优缓存节点选择算法来证明所提算法的实际适用性。最后仿真实验结果验证了本文提出的最优缓存节点选择算法和主从节点协作缓存算法的有效性。

关 键 词：D2D，缓存，节点选择，协作缓存

ABSTRACT

With the development of information technology and mobile devices, especially the popularization of 4G and the commercial use of 5G, network resources such as video have shown an explosive growth. Large amount of network traffic, especially in the peak period, will cause network backhaul congestion and make users' request delay increase. How to solve the problem of network congestion is the focus of network structure optimization.

In order to alleviate network congestion and reduce request delay, D2D caching technology will be an important part of modern communication network. People always browse the Internet for things they are interested in. D2D caching technology can cache the content that people are interested in. When people request the cached content, they can get it directly through D2D communication without requesting to the base station. Since different people have different interest points, and people's interest points will change with time and the emergence of social hot spots, the selection of cache nodes and how to effectively determine the cache content have become an urgent problem. The present research is mainly based on the user's device capability and historical communication record to decide the cache device. In fact, the distribution of users' requests around the cache node is constantly changing and the amount of historical data is not sufficient, which make the selection of cache nodes and cached content keep changing.

In this paper, an optimal cache node selection algorithm based on virtual delay is proposed, which is based on the multi-armed bandit model and obtains the optimal cache decision according to the interest differences of users. Each candidate user has the opportunity to become a cache node. The algorithm selects the candidate user with the least system delay as the cache node. In practice, there are multiple cache nodes in a given region. Under the condition of limited cache node space, how to improve the cache hit ratio of requesters is an important problem. The current research focuses on the way to collaborate between multiple cache nodes. Because the cache capability of each cache node and the request distribution of surrounding users are different, the service capability of each cache node is different, so it is necessary to effectively distinguish the service capability of each cache node, and then make full use of the service capability of each cache node for maximizing the overall benefit. Therefore, this paper proposes a master-slave node cooperative caching algorithm. Based on

the optimal cache node selection algorithm, this algorithm makes full use of the collaboration between cache nodes with strong service capability, and adjusts the cache decision of each cache node to minimize the overall request delay. In addition, considering the fact that most users have mobility, this paper tests the proposed optimal cache node selection algorithm based on virtual delay in a moving scenario to prove the applicability of the proposed algorithm. Finally, simulation results verify the effectiveness of the proposed optimal cache node selection algorithm and master-slave collaborative cache algorithm.

Keywords: D2D, Cache, Node selection, Cooperative cache

插图索引

表格索引

符号对照表

符号	符号名称
A_t	时间 t 采取的动作
R_t	时间 t 采取 d 动作获得的奖励
$q_*(a)$	给定动作 a 的预期奖励
$Q_t(a)$	动作 a 的估计值
ε	从动作空间中随机选择动作的概率
$P(k)$	Softmax 函数转化的一系列概率值
$N_t(a)$	在时间 t 之前选择动作 a 的次数
U	用户组
N	用户组内用户的个数
$dis_{m,n}$	用户 u_m 和用户 u_n 之间的距离
r	两用户进行 D2D 通信的最大距离
$R_{\mathrm{n},m}^{d2d}$	用户 u_m 和 u_n 之间的 D2D 连接传输速率
$R_{0,m}^{\mathrm{bs}}$	用户 u_m 和基站之间的连接传输速率
W	传输链路的带宽
P_n	用户 u_n 的传输功率
$h_{n,m}^2$	用户 u_n 与用户 u_m 之间的链路系数
$r_{n,m}^{-\alpha}$	用户 u_n 和用户 u_m 之间的路径损耗
$r_{n,m}$	用户 u_n 和用户 u_m 之间的物理距离
α	路径损耗指数
σ^2	每个用户的噪声功率
$d_{m,d2d}$	用户 u_m 通过 D2D 连接接收到完整的内容需要花费的时长
$d_{m,bs}$	用户 u_m 从 BS 接收到完整的内容需要花费的时长
F	服务器的文件库
HU	候选用户集
C	每一个候选用户的设备的存储能力
γ_i	兴趣偏好因子
$p^{i,k}$	文件请求概率
L	缓存节点用户的缓存文件集合

$I(t, p^{u_i, f_k})$	用户 u_i 是否缓存了文件 f_k
$D(t)$	当前时隙下整个关系组中的用户所经历的整体时延
$D(u_i, t)$	用户 u_i 在当前时隙下经历的时延
$Reward(t)$	当前时隙下的奖赏
$In(r)$	两用户之间的亲密程度
$V(I(t, p^{u_i, f_k}))$	考虑用户亲密度的奖赏
v_m^t	相对移动速度
θ	运动的方向角度
MU	主区域特有用户集合
CU	交叉区域用户集合
RU	从节点剩余区域用户集合
$d_{m,h}^t$	用户 u_m 与候选用户 u_h 之间的距离
$de_{m,h}^t$	移动模式下用户 u_m 与候选用户 u_h 之间的时延

缩略语对照表

缩略语	英文全称	中文对照
2G	The 2th Generation	第二代移动通信
4G	The 4th Generation	第四代移动通信
5G	The 5th Generation	第五代移动通信
BS	base station	基站
D2D	device-to-device	终端直通
LFU	Least Frequently Used	最不经常使用
LRU	Least Recently Used	最近最少使用
MAB	multi-armed bandit	多臂赌博机
SBS	small base station	小基站

目录

第一章　　绪论

1.1 研究背景与意义

通信技术及相关产业的发展带动着通信设备的快速变化，从 2G 时代到已经商用的 5G，人们对移动通信设备产生了越来越高的依赖。工信部信息显示，2019 年 1-11 月全国 4G 用户达到了 12.7551 亿用户，比 2018 年末净增 8.6%，移动互联网用户 13.0847 亿用户，比 2018 年末净增 2.6%。以上数据表明通信设备几乎覆盖了社会的每一个角落。随着智能设备和网络的普及以及人们对网络的依赖逐渐增强，网络冲浪已经成为人们生活密不可分的一部分。随着 5G 技术的商用[1]，人们对于高清快速的多媒体业务需求变得更大，对网络的请求体验变得更加敏感。网络中的多媒体内容急剧上升对现有的网络结构提出了更高的要求。大规模的请求会造成网络的拥挤，增加用户的请求延迟，降低人们的上网体验。在海量的移动智能设备与日俱增和用户特定需求的新兴通信业务呈现爆发之势的场景下，有效的提高蜂窝系统容量、减缓网络拥塞、提高频谱利用率和终端用户请求满意度是研究的热点问题。

终端直通(device-to-device, D2D）技术[2]被认为是一个有效的缓解网络拥塞的优化手段。D2D 通信技术已经是第五代移动通信(5th-Generation,5G)的关键技术[3]之一。D2D 通信技术是指在系统控制的情况下，通信设备不需要基站转发数据直接建立通信连接交换数据的技术。该技术既可以在较近距离的情况下以较小的功率进行快速数据传输，也可以复用移动通信网络的频谱资源来提高频谱的利用率。另外，通信双方的服务质量也都有较好的保证以及更大的灵活性。D2D 通信[4]的应用场景之一是基于周边用户的社交应用[5]，用户可以通过设备发现功能找到周围的终端设备进行终端之间的数据传输[6]。其他的应用场景也可以是定向广告投放、应急通信以及物联网增强等领域。

本文主要研究的是基于周边用户的社交应用场景[7][8]。在用户请求高峰期的场景下，比如下班后的休息时段，网络拥挤成为了一大问题。现阶段的优化手段包括增加 SBS(Small BS)等手段来增加网络容量，但是这会带来相应的基础设施建设，新增建设成本和后期的人工维护成本。此外，5G 的商用[9]带来了更大的数据流量，集中式的基站请求模式和有限的回程流量会导致严重的网络拥挤，因此，D2D 通信作为新的通信模式在新时代的背景下会发挥更大的优势。

D2D 通信技术和缓存技术[10]相结合可以更加有效的缓解回程流量拥挤的问题。缓存技术目前是一个应用广泛的技术。通过在基站或者小基站部署相应的缓存设备以存储用户可能感兴趣的内容，在用户请求来临的时候可以不通过基站就可以获取到相

应的内容。这对改善服务端应对大规模请求具有显著的效果。随着制造工艺的发展，现在的移动智能设备(移动手机和笔记本电脑等设备)具有越来越大的存储空间，而对于某些用户来说，他们的设备的剩余存储资源没有得到有效的利用。为了能够让用户自愿利用自己的存储设备来为周围用户服务，运营商可以提供一定的激励[11][12]。在文献[4]和[5]中作者研究了用户如何在激励的条件下进行 D2D 相关技术[13]的研究。用户在一定的激励下利用自己遗留的空间来缓存一些热点内容。在用户请求热点内容时便可以直接从缓存设备上得到内容，这样便可以不经过蜂窝网络就可以得到用户想要的内容，缓解了网络回程拥挤。但是在有限的存储空间下，如何使得缓存命中率最大化是当下的一大挑战。缓存命中率依赖于缓存的内容，缓存命中率越大，周围用户的传输时延就会越低。因此，如何决定缓存的内容成为了当下研究的热点。在缓存决策时，周边用户的兴趣时变性、终端设备有限的存储空间、周围用户的时变的请求频率以及不可预知的网络热点等因素决定了并不能使用一成不变的内容评估方法来决定存储哪些内容，应用机器学习的手段来做缓存方面的研究展现了强大的生命力。其中强化学习的表现尤为突出。文献[14,15,18]中均使用了强化学习手段[14][15]来做相应的缓存决策[16][17]，都取得了较好的效果。强化学习[18]不同于监督学习[19]和非监督学习，它在不要求预先给定任何数据的条件下，智能体以试错的方式与环境进行交互并获得奖赏来指导下一步的行动，整体的目标是使得获得的奖赏达到最大。在众多因素的影响下，通过多次的迭代学习，智能体总能找到一种最优的行动策略，这使得我们在做缓存节点选择和缓存决策时有了很重要的依据。因此，现阶段在做缓存系统[20]时大多会使用强化学习的手段来做内容缓存[21]。这种方式使得缓存设备在历史记录较少的情况下根据当下用户实际请求缓存用户请求的内容，并根据相应算法进行缓存的更新。

1.2 国内外研究现状

通过不同缓存机制和 D2D 技术[22][23]的结合来提升系统性能的方法已经被广大学者研究应用。在不同缓存机制下，时延值、命中率等衡量缓存机制有效性的指标都得到一定程度的改善，但相应的也存在着一些问题。

文献[15]和[23]证明了蜂窝网络中的用户可以通过 D2D 直接交换信息，而不需要基站的帮助。文献[20]和[21]证明了 D2D 主动缓存在节省回程资源和用户满意度方面为系统带来了良好的收益。文献[10]和[11]提出了基于社会感知的激励缓存模型，促进节点主动为其他节点缓存内容。文献[6]使用与物理距离相关的社会距离模型，该模型使用随机和确定性缓存策略来获得平均下载延迟性能。在文献[6,10,11]中均提到用户的社会关系是研究 D2D 缓存的一个重要因素。文献[4]中提出了一种混合缓存策略框架。考虑到用户的自私性，利用随机几何知识将用户的位置建模为高斯泊松过程，

并通过用户之间的协作缓存来增加收益。文献[40]作者采用了在多个基站之间直接联合学习缓存决策的策略，而不需要首先评估用户对内容的偏好。文献[14]提出了一种增强学习模型，用于在移动 D2D 网络中未知内容流行度的情况下，缓存单用户和双用户之间的决策。但是文献[4,14,40]并不考虑用户之间的社会关系，这将对实际应用产生重要的影响。本文在之前研究成果的基础上，提出了用户亲密关系组的概念。用户关系组包括用户的同事关系、同学关系、邻居关系、家庭关系等。这些关系组在实际生活中是存在的，用户之间存在一定的信任值，这是 D2D 网络的重要组成部分。将整个组作为一个整体，使用 D2D 缓存模型进行信息共享符合实际。在用户亲密关系组中，如何设计一个有效的缓存机制是一个值得关注的问题。为了在社会福利和网络请求延迟方面获得更好的性能，文献[5,6]提出了一种分布式缓存系统，该系统使用匹配理论来决定用户应该缓存哪些文件。在文献[5]中有一个缓存用户选择方法。但是，该方法根据用户的意愿和兴趣相似度选择用户作为缓存节点。而且，被选择的用户需要放弃他们的兴趣点。以这种方式选择的缓存节点，必然会失去一部分周围用户的兴趣偏好，所选择的用户可能不是能够为整体带来最大收益的用户。因此，本文使用一种基于虚拟时延的最优缓存节点选择算法来评估候选集的用户，并根据他们带来的总体效益进行排序。但是，用户的自私性(比如电池没电)将使系统变得脆弱，而且单个用户存储的内容数量是有限的。每个单独进行缓存决策的节点不利于提高缓存命中率。根据文献[4,14,40]，协作缓存可以给系统带来很好的性能提升。它不仅可以减少用户的自私性对系统的影响，还可以提高存储容量和内容多样性。但是上述文献忽略了一点，即由于不同的用户缓存行为(在线时间、设备容量等)，它们给系统带来的好处是不同的。因此，本文根据用户对系统的缓存服务能力对用户进行排序，选取缓存服务能力强的用户参与协作缓存。基于用户自身的缓存能力、系统的可持续性和内容的多样性，本文提出了一个主从节点协作缓存模型。

文献[20]中使用 MAB 框架来估计文件的受欢迎程度，然后根据受欢迎程度缓存相关文件。文献[24]中提出了一种基于迁移学习的估计方法。它提出通过参数族分布对内容流行度模型进行建模。文献[24]中使用的策略是，如果缓存的内容流行度未知则首先评估内容流行度。文献[20,24]采用了强化学习的方法来确定缓存的内容，但这里采用的方法是先评估内容的流行度，然后再做出缓存决策。但是，文件可能在每个时隙被缓存而且周围的用户有不同和动态的兴趣点。随着时间的推移，首先估计内容的流行度的方法虽然会取得一定收益，但是其缓存的文件却不是最优的。更重要的是该算法复杂度较高，收敛速度较慢。因此，本文采用的方式是基于未知内容流行度通过环境反馈直接进行缓存决策。

文献[43]中利用了移动的大型缓存设备来帮助用户缓存内容，这对终端用户的移动具有一定的参考意义。实际中用户具有移动性，而其携带的缓存设备也会随之移动。

因此，研究用户的移动性对系统的实际应用具有一定的意义。但并不能与文献[43]完全等效，文献[43]是在确定缓存设备的情况下做出缓存决策，而本文的目标是在做缓存节点选择时就考虑用户的移动性。

1.3 论文的研究工作

本文主要根据已有的研究文献[6,10,11]中对用户亲密度和用户距离的频繁使用，并结合现实生活场景中的人际关系，提出了用户亲密关系组的概念。用户亲密关系组指由家庭关系、邻居关系、同学关系、同事关系等相关人员组成的群组。在此基础上，为使整个关系组请求延迟达到最小，对关系组内所有用户的兴趣偏好和请求意愿进行整合，执行缓存节点的选择策略。在缓存节点选定的情况下，在未知内容流行度的情况下基于多臂赌博机模型直接进行缓存内容的替换，最终学习到缓存决策内容[24]。考虑到用户的自私性和协作缓存的有效性，本文根据用户的缓存服务能力不同的特点，提出了主从节点协作缓存模型。在实际场景中，关系组内用户多处在移动场景中，因此本论文在之前的研究的基础上测试了用户移动性条件下缓存节点选择策略。本文的研究工作具体如下：

第一，提出了用户亲密关系组概念，在此基础上，考虑关系组内所有用户的兴趣偏好，为使整体请求延迟达到最小，提出了基于虚拟时延的最优缓存节点选择算法，并通过多臂赌博机模型进行缓存决策。在缓存节点选择时，主要对多个候选节点进行缓存服务能力排序，得到缓存节点的候选序列。

第二，提出了基于实时 MAB 算法的主从节点协作缓存算法，该算法作用范围为整个关系组。当关系组内出现多个缓存节点时，参考缓存节点候选序列，希望整体用户请求延迟达到最小，则需要使缓存空间的内容多样性提高，尽可能的缓存更多的非重复性内容。

第三，研究了用户移动条件下基于用户亲密关系组的缓存节点选择策略和缓存决策。主要考虑普通请求用户的移动和候选用户的移动对缓存节点的选择和缓存内容的影响。

1.4 论文的结构安排

本论文的组织结构安排如下：

第一章，介绍了本文的研究背景与研究意义。包括现阶段 D2D 通信领域的优势与发展趋势，并介绍了缓存技术、强化学习与 D2D 技术的结合带来的相关优势与问题。然后介绍了国内外学者对缓存替换策略和缓存节点选择策略的研究进展，提出了对已有研究的看法，并根据实际生活场景提出了改善的缓存节点选择策略和学习策略

以及进一步的协作模型。

第二章，首先介绍了 D2D 通信的相关技术--设备发现和模式选择，主要考虑的是用户之间在基站的帮助下完成连接建立和如何传输数据的过程。这两种技术在整个模型的应用中是最先需要使用的技术。接下来介绍了强化学习相关的发展历史，并详细解释了强化学习的主要思想和模型，接着介绍了强化学习中有关多臂赌博机模型(multi-armed bandit, MAB)的几种经典算法。在本章的最后介绍了移动边缘缓存技术。

第三章，主要介绍了跟系统模型有关的通信模型、缓存模型和奖赏模型。并给出了基于实时 MAB 算法的缓存更新算法。接着详细介绍了缓存节点选择策略，该策略主要使用了虚拟时延的思路。在缓存节点选择策略的基础上，介绍了主从节点协作缓存模型，该模型考虑了整个用户亲密关系组的兴趣偏好，使得整体时延达到更低的水平。并在此基础上进一步的研究了用户移动条件下对缓存节点的选择策略和缓存决策的影响。

第四章，主要对第三章提出的模型和算法进行仿真验证，主要分为两部分。第一部分是用户静止条件下对所提模型和算法的仿真结果与分析，第二部分是用户移动条件下对所提模型和算法的仿真和分析。

第五章，总结所做的工作，并根据已有结果提出了未来进一步研究的方向。

第二章　D2D 通信技术与强化学习缓存算法

为了优化网络结构，减轻高峰期的网络拥挤，D2D 通信技术与缓存技术的结合展现出了良好的应用前景。强化学习的发展使得缓存的内容更加精确化，强大的内容指向性进一步提高了请求命中率，促进了 D2D 与缓存技术的发展。本章将详细介绍本文所用到的 D2D 通信的关键技术、缓存技术和强化学习算法。

2.1 D2D 通信

传统的蜂窝通信系统以基站为核心对周边区域实现小区覆盖，但是这种方式导致基站、小基站以及缓存节点的位置固定，从网络结构上看缺乏一定的灵活性。随着 4G 的普及以及 5G 的商用，不断增多的无线多媒体业务导致传统的基站服务模式无法满足海量用户的无线业务需求。D2D 通信[25]不经过基站转发就可直接进行数据传输[26]，可以对网络边缘进行拓展，创新接入方式。D2D 通信技术已经是第五代移动通信的关键技术之一。由于不需要基站的转发，D2D 技术以其短距离高质量的直接通信，实现了高速率的数据传输和较低的网络时延，有效的缓解了网络拥挤，减轻了核心网络负担。海量用户终端的广泛分布改善了小区覆盖模式，在现阶段频谱资源有限的情况下，有效提高了网络容量和频谱利用率。

2.1.1 应用场景

在无线接入层面，按照蜂窝网络的覆盖范围划分，可以将 D2D 技术应用在 3 种场景下：蜂窝网络控制下的 D2D 通信、蜂窝网络辅助控制下的 D2D 通信、不受蜂窝网络控制的 D2D 通信。

- 蜂窝网络控制下的 D2D 通信：在这种场景下，基站负责通信的全过程。它会首先发现需要进行 D2D 通信的设备，然后建立逻辑连接，接着基站会控制 D2D 设备的资源分配[27]，进行一系列的资源调度和干扰管理，用户可以得到质量较高的数据传输。
- 蜂窝网络辅助控制下的 D2D 通信：在这种场景下，基站负责的事情比第一种较小，网络复杂度有明显下降，基站只需要引导需要通信的双方建立连接，并不参与资源调度等过程。
- 不受蜂窝网络控制的 D2D 通信：这种场景则完全不需要基站，即不在任何蜂窝网络覆盖下，通信设备双方在蜂窝网络处于瘫痪的状态下，通过单跳或者多跳的方式可以进行直接通信或者接入网络。

2.1.2 D2D 的关键技术

两个设备如果要成功建立通信，必然需要先进行发现彼此然后建立通信连接。接下来主要介绍 D2D 中非常关键的技术：设备发现[28]和模式选择[29][30]。

1、设备发现[31]

通信双方知道彼此的存在是建立通信的前提条件。对于 D2D 通信的双方，不像基站等大型设备具有固定的位置信息，通信一方如何知道另一方在什么位置等问题已经被众多研究者所研究。大致可以分为两类：基于基站的辅助设备发现、基于用户设备的自主发现。

基于基站的辅助设备发现指的是当有用户需要进行 D2D 通信的请求时，它会接收请求信息，然后基站会根据请求终端的位置信息按照其可以继续通信的范围测量其周围终端的信道状态，若其周围的设备愿意进行 D2D 通信且通信条件达到要求，基站则会帮助双方建立连接。基站在整个过程全程参与，会消耗不少的的能量和造成一定的资源开销。

基于用户设备的自主发现的方法是在建立 D2D 通信时并不需要基站的参与，而是由 D2D 设备终端自主完成连接过程。在这个过程中，希望进行 D2D 通信的用户需要对外广播自己的通信需求。一般情况下，由于设备有限的能力，广播范围并不会太大。在该范围内的用户如果接收到该请求并且愿意进行 D2D 通信的情况下，就可以进行 D2D 连接建立。但是这里的问题是要求用户终端需要一直或者不间断的进行请求连接信息的发送，对自身设备的能力有很高的要求，在这一点上，如何有效的建立 D2D 设备的自主发现连接也是一个有趣的问题，使得广大感兴趣的研究者进行了深入研究。

2、模式选择

D2D 通信双方传输数据必然要占据相关频谱资源，在蜂窝网络场景下，终端设备应该根据实际情况灵活选择自己的工作模式，来最大化利用频谱资源。在 D2D 用户和蜂窝网用户同时存在的情况下，主要以下三种模式可供选择：

- 蜂窝模式[32]：该模式下 D2D 用户和蜂窝用户在模式使用上并没有什么特别的不同，D2D 用户需要传输的信息由基站转发，基站为 D2D 用户分配相互正交的上下行频谱资源，在这种情况下，D2D 用户和蜂窝网用户之间并不存在干扰。
- 专用模式：在该模式下，主要是给 D2D 用户分配一些还没有被使用的频谱资源。但是这会带来新的问题，在频谱资源有限的前提下，一部分资源给了 D2D 用户，相应的分配给蜂窝用户的资源就会变少。
- 复用模式：D2D 用户和蜂窝用户使用相同的频谱资源进行数据传输，这种情况一般发生在基站已经没有可用的频谱资源分配给 D2D 用户。虽然 D2D 用户和蜂窝

用户之间会造成干扰，但也会相应的带来频谱资源利用率的上升。如何处理干扰问题也是一个研究的热点。

2.2 强化学习

计算能力的发展催动了机器学习技术在近 10 年内的飞速发展。机器学习[33]主要研究的是智能体的学习过程以及通过学习改善处理某些问题的能力。所谓学习就是智能体在环境中不断通过一些实践来训练自己达到能够智能处理问题的能力。机器学习有多个方法，强化学习就是其中一个重要方法。强化学习的指导思想与行为心理学研究有很强的关联。1911 年 Thorndike 提出了效用法则，该法则指出，在一定的情境下如果想让某动物与某种动作加强联系，就让动物做出一些感到舒服的行为。这样一来，当相同的情景再次出现时，动物会做出同样的行为，也就是说，该行为与此情景加强了联系。相反，如果不想让动物在某些场景下不做出某些动作时，就让动物在做出该动作时让动物感觉不舒服，以此来减弱该动作与此情景的联系。在实际生活中，例如在训练员做出转圈手势时，海洋馆的海豚若做出转圈的动作时，训练员会给它鱼吃作为奖励。海豚转圈这个行为和训练员做出转圈手势这个情景的联系就得到了加强，有鱼吃的奖励使得海豚记住转圈这个行为。在特定的场景下，那些得到奖励的行为会被不断加强，那些得到惩罚的行为会被不断减弱。在这种模式下，生物可以从不同的行为获得的奖励或惩罚学会在此模式下应该选择什么样的行为才能达到训练者的最想要的行为。强化学习的核心指导思想和这种生物模式是一样的。通过在给定的场景下不断的尝试不同的行为获得不同的奖赏或惩罚来选择最合适的行为。强化学习不像其他的机器学习的方法，告诉智能体去选择哪种行为而是通过不断的尝试自主发现可以产生最大奖赏的行为并选择它。

强化学习被 Sutton 描述为为了获得最大的回报而不断的尝试去匹配最佳的状态和动作。这使得强化学习在机器人交互、策略博弈、对话系统、传媒和广告、控制系统等领域有了很多的应用。比如由谷歌旗下公司 DeepMind 公司研发的 AlphaGo 第一次战胜了围棋世界冠军，而后来应用了强化学习手段的 AlphaGo Zero 战胜了 AlphaGo，使得研究者门对强化学习产生了更大的兴趣。之后在游戏领域的 dota2 中强化学习战胜了职业战队。越来越多的案例证明了强化学习的强大的学习能力。

2.2.1 强化学习的发展背景

强化学习发展至今也不过几十年的历史。1953 年，应用数学家 Richard Bellman 提出的动态规划数学理论[34]和方法中的贝尔曼条件(Bellman condition)是强化学习的基础之一。1957 年，Richard Bellman 提出了马尔可夫决策过程，并且使用了类似强

化学习的试错机制来求解该方法。正确的理解马尔可夫决策过程对学习强化学习至关重要，以至于马尔可夫决策过程成为定义强化学习问题的普遍形式。20 世纪 60 年代，强化学习概念开始在工程文献中出现。1963 年，Andreae 开发出 STeLLA 系统，可以通过与环境交互进行试错学习。Donald Michie 描述了 MENACE——一种试错学习系统。1965 年在控制理论中也出现了使用奖罚的手段进行学习的思想。这是早期简单的探索性研究，试错机制也成为了一种较为通用的求解强化学习问题的手段。1989 年，Watkins 提出来的 Q 学习[35]使得强化学习得到完善并促进了其进一步的发展。他还证明了在一定的条件下强化学习是收敛的。还有时间差分法[36]、自适应动态规划[37]、部分可观测马尔可夫决策过程等等都是强化学习进一步发展的基石。直至 21 世纪出现了深度 Q 学习[38]、确定性策略梯度学习、双 Q-learning 深度强化学习等强化学习的经典算法。

2.2.2 强化学习模型

强化学习，又称增强学习，是机器学习领域的中重要分支。现阶段在推荐系统设计、机器人交互、游戏等领域具有较多的应用。它最主要的目标是使得智能体收获最大的奖赏，在信息有限的条件下，它的行为模式主要参考的是做出的行为对环境的影响以及环境给出的反馈。

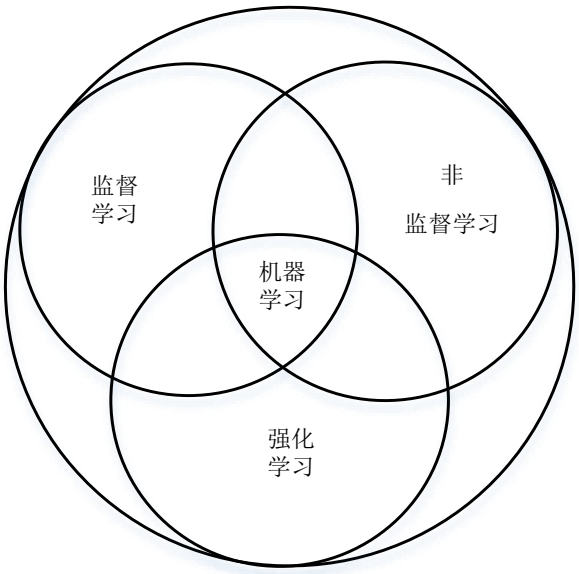

图 2.1 强化学习与机器学习其他分支的关系

如图 2.1，监督学习是在已经有标签的训练集的基础上进行学习训练，在该训练集中的每一个样本都由特征和标签组成，每一个特征都可以视为对情景的描述，每一个标签可以看作是正确的行为，监督学习在分析此数据集之后，产生一种适合此类数据的推理功能，使得该训练模型可以对其他的案例进行结果推理。但是监督学习并不

适合在交互的场景下进行学习，因为交互问题有一个很显著的特点，其获得期望行为的案例很不实际，智能体从自己已有的行为经历中进行学习，这个经历有时并非是理想的行为。这样的场景使用强化学习就非常合适，因为强化学习是根据已有的训练信息来学习而不是根据正确的行为来训练。

无监督学习的很显著的特征是它的数据集并没有被标记，这在现实生活中有很多场景，比如，有些数据缺乏足够的先验知识无法进行标注，或者是人工标注的成本很高等。这类数据集中的样本数据的类别并不知道，需要根据样本之间的共同特性来对样本进行一定程度的分类使得同一类的样本差距最小，使得不同类的样本间差距最大。非监督学习是让智能体自己学习怎样去做事情而并不告诉智能体如何做。非监督学习和强化的共同点在于强化学习利用的并不是采取正确的行动，强化学习的目的是最大化奖赏，而非监督学习的目的是从一对杂乱无章的数据中找到其内部隐含的结构。

强化学习的特别之处在于：仅有奖赏信号并无监督者，环境的反馈并不是立即生成而是有延后性，时间对于强化学习的意义非凡，智能体的行为会对以后的行为选择产生重大影响。其主要工作模型如图 2.2。

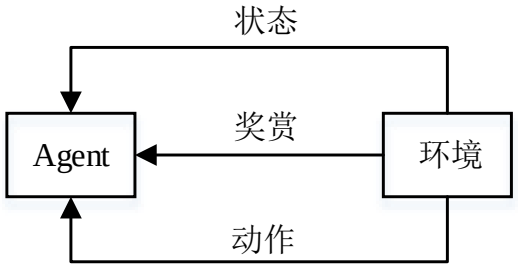

图 2.2　强化学习模型

强化学习将学习的过程看作是一个试错的反馈过程，在探索和利用之间取得一个平衡，当智能体对环境产生一个动作，环境会在当前动作的效应下使得自身的状态发生变化，此时环境中由于接受智能体的动作产生的结果会给智能体一个反馈，对于在这种反馈中做出的下一个选择动作，我们希望看到的是能使得正反馈的概率增大。这样就可以使得我们得到正奖赏的概率增大，通过一段时间的学习之后，达到我们想要的最终目标。

在模型设计上，需要考虑以下因素：1、状态空间和动作空间；2、建立什么样的信号和如何通过学习修正不同的状态和动作；3、选择下一动作的依据；

从图中可以看出，其主要由 5 个因素组成。分别是智能体、动作、环境、状态和奖赏。

- 智能体是一个可以选择动作的智能个体，对应在现实生活中就是我们自己。在工业领域，可以是一个无人驾驶的汽车或者飞机、一个自助服务助手、一个智能语音音箱等等。
- 动作代表的是智能体可以采取的动作的集合。智能体做出的动作对我们看来是很简单，但是需要明白的是这个动作是从可能的动作选择中筛选出来的。在游戏领域此动作结合可以是向前走、静止不动、向后走、向上跑、向下跑等。在股票买卖中，动作集合可能包括买入或者卖出股票等。
- 环境是智能体所处的周边环境。这个环境可以让智能体处理各种行动以及决定接下来可能发生的结果。在此环境中，将接收的智能体的状态和动作作为输入，将智能体的奖励和下一步的状态作为输出。
- 状态是智能体所处的具体实时状态，它是环境返回的当前的具体形势。它能够将智能体和其他重要的事物关联起来，例如工具、敌人和或者奖励。
- 奖励是我们对智能体的行动好坏的一种反馈。比如，在游戏领域，若是人物捡到了金币或者消灭了怪物，那它就会获得一定的分数奖励。在既定的状态下，智能体和环境交互的方式是智能体向环境发出一个动作，然后环境受到该动作的影响会给智能体返回一个新的状态和奖励。这个奖励可以是瞬间反馈的也可以是延后反馈的。

2.2.3 现阶段算法

本小节主要介绍几种强化算法，包括多臂赌博机算法、$\varepsilon-greedy$ 算法、softmax 算法和 UCB 算法。同时介绍一下增量实现方法。这几种算法的目的相同，但是采用了不同的实现方式，都是在自己的考虑条件下来实现收益最大、与实际更相符的愿景。

1、多臂赌博机算法

多臂赌博机算法来源于赌博机，当我们在玩这款机器时，不免想让自己获得更多的金币，这时候需要一种机制来决策自己接下来的动作。考虑赌博机模型，它有多个拉杆，每一次只能拉下一个摇杆，每一个摇杆拉下后都有一定的概率给出一定的奖励，在概率分布未知的情况下，我们希望能够获取最大的金币量，一个行的通的方案是通过反复的拉动摇杆，把拉动摇杆的动作固定在最佳的摇杆上来最大化金币量。对应在强化学习模型中，每个动作选择就像一个赌博机的摇杆，奖励是拉下该摇杆可以给出的金币量。该模型可以概括多数同类问题。

将上述问题建模为数学问题就是我们有 k 个动作，每个动作都有一个预期或者平均的奖励值，将它称之为该动作的价值。用 A_t 表示在时间 t 采取的动作，用 R_t 表示该动作获得的奖励。于是对任一的动作 a 的价值使用 $q_*(a)$ 定义为在给定动作 a 的预期奖励：

$$q_*(a) = \mathrm{E}[R_t \mid A_t = a] \tag{2-1}$$

若是我们知道每个动作 a 的价值，只要选择价值最高的那一个动作就可以了，该问题就可以很轻易的得到解决。但是实际中我们仅仅知道某一动作的估计值 $Q_t(a)$，并不知道某一动作对应的真实价值。用 $Q_t(a)$ 表示在时间 t 下动作 a 的估计值，我们的目标是能够使得 $Q_t(a)$ 尽可能的接近 $q_*(a)$。

在上述模型中，如果我们一直保持对动作的估计值，在某一时隙中我们可以得到一个最大估计值的动作，这种做法称为贪心策略。我们在做每个动作选择的时候利用了我们当前对每个动作的价值估计信息，若我们采取了一个不在估计信息范围内的动作，这其实是一种探索策略。探索的结果是我们可以获得对当前动作的价值估计信息。以此来丰富我们的的信息库，以合理的做出下一个动作选择来不断接近我们的奖励最大化目标。利用和探索策略给了我们短期和长期的奖励保证。在短期内，通过利用已有的对动作价值的估计信息使每一次的动作选择带来的奖励达到最大。在长期内，通过探索策略使得我们了解了更多的未知的可能存在的更大价值的动作选择。例如，在当前已知的动作估计信息的基础上，贪心策略使得我们可以选择价值较高的动作，但是其他的未知的动作选择存在着很大的不确定性，但可能会有一个比当前贪心策略选择的动作的价值更高，这虽然在短期内可能获得奖励并不会很高，但是从长远的角度来看可以多次利用价值更高的动作行为来达到我们的奖励最大化目标。

因此，如何在利用和探索之间寻找一种平衡是多臂赌博机模型需要认真考虑的问题。接下来介绍几种平衡方法。

2、$\varepsilon - greedy$ 算法

根据上一小节的分析，要想在当前估计信息范围内获得最大奖励，在每个时隙只需要选择具有最高价值回报的动作即可。也就是说在整个动作空间中，当前动作的选择应该满足下式：

$$A_t = \arg\max_a Q_t(a) \tag{2-2}$$

式(2-2)表达的是当前动作估计价值最大的动作选择。而 $Q_t(a)$ 代表的是选择动作 a 带来的的平均奖励，也就是，在当前时隙之前采取该动作获得的所有奖励值与采取该动作的次数的比值。

$$Q_t(a) = \frac{\text{在t之前采取a动作的奖励总和}}{\text{在t之前采取a动作的次数}} = \frac{\sum_{i=1}^{t-1} R_i \bullet \mathrm{I}_{A_i=a}}{\sum_{i=1}^{t-1} \mathrm{I}_{A_i=a}} \tag{2-3}$$

其中 I 代表指示函数，指示的是当前动作选择的是 a 时该值为 1，否则为 0。当采

取该动作的次数为 0 时，将$Q_t(a)$设为 0。当采取该动作的次数为无限次时，根据大数定律，$Q_t(a)$将收敛于$q_*(a)$。

贪心算法总是在已有的认知知识下选择使得即时奖励最大的价值动作，并不会理睬那些价值偏低的动作。但是贪心算法只是在利用已有的知识，为了在探索和利用之间找到平衡，一种简单的做法是在每个一段时间，以一个较小的概率ε，从当前所有的动作空间中随机选择，这种做法称为$\varepsilon-greedy$算法。在足够多的选择次数内，该动作空间的所有的动作均会被采取，在贪心策略下选择的具有最优价值的动作概率会近乎确定下来以确保所有的$Q_t(a)$无限接近于$q_*(a)$。$\varepsilon-greedy$算法主要流程如图 2.3。

ε-greedy算法
输入：摇臂数K; 奖赏函数R; 尝试次数T; 探索概率ε. 过程: Step1:初始化奖赏值、每个摇臂对应的平均奖赏、 每个摇臂的摇动次数为0; Step2:对于每一时隙，如果随机函数rand() < ε， 则从K个摇臂中以均匀分布选取；否则，选择 平均奖赏值最大的那个摇臂; Step3:记录每次选取的摇臂对应的奖赏并计算累积奖赏， 采用增量实现更新采取的摇臂对应的平均奖赏; 将该摇臂的摇动次数加1; Step4:重复Step2直到设定的尝试次数T; 输出：累积奖赏、平均奖赏值;

图 2.3 $\varepsilon-greedy$算法

3、Softmax 算法

Softmax 算法基于当前每个动作的平均奖赏值来对探索和利用进行折中。这里首先介绍一下 softmax 函数。

$$P(k)=\frac{e^{\frac{Q(k)}{\tau}}}{\sum_{i=1}^{K}e^{\frac{Q(i)}{\tau}}} \tag{2-4}$$

该函数将不同的值转化为一系列概率值，若各动作的平均奖赏相当，对应的选取各动作的概率也相当；如果某些概率的平均奖赏明显高于其他概率的奖赏，则它们被选取的概率也明显高。在该函数中，$Q(i)$记录的是当前动作的平均奖赏；当$\tau>0$时，T越小平均奖赏高的动作被选取的概率就会越高，在τ约等于 0 时该函数的效果和贪

心算法中的仅进行利用的效果差不多，在τ趋于无穷大时该函数的效果和贪心算法中仅进行探索的效果差不多。在贪心算法中，ε的值可以由我们自己设定，但在 Softmax 算法中动作空间中各动作的概率分布满足 Boltzmann 分布，也就是 softmax 函数。Softmax 算法主要流程如图 2.4。

Softmax算法

输入：摇臂数K;
　　奖赏函数R;
　　尝试次数T;
　　温度参数τ.
过程：
Step1:初始化奖赏值、每个摇臂对应的平均奖赏、
　　每个摇臂的摇动次数为0;
Step2:对于每一时隙，根据式(2-4)从K个摇臂中随机选取
　　摇臂;
Step3:记录每次选取的摇臂对应的奖赏并计算累积奖赏，
　　采用增量实现更新采取的摇臂对应的平均奖赏;
　　将该摇臂的摇动次数加1;
Step4:重复Step2直到设定的尝试次数T;
输出：累积奖赏、平均奖赏值;

图 2.4　Softmax 算法

4、UCB 算法

由于对于动作空间中的动作价值的估计存在一定的不确定性，因此贪心算法中的探索行为是很有必要的，但是也有另一种方法在实际应用中比使用探索更好。因为在ε贪心行动中会有一些本身并不具有贪心价值的动作进入了探索流程，这使得那些具有贪心价值的动作并没有得到充分的挖掘。UCB 算法使用了这样一种方法，它根据实际中具有最优价值的潜力来选择那些不在贪心范围的动作，同时考虑到了这些动作的估计值和最大值的估计值的接近程度以及这些估计中的不确定性。

$$A_t = \arg\max_a \left[Q_t(a) + c\sqrt{\frac{\ln t}{N_t(a)}} \right] \tag{2-5}$$

其中$\ln t$表示t的自然对数(e≈2.71828 必须提高到等于 t 的数量)，$N_t(a)$ 表示在时间t之前选择动作a的次数，数字$c>0$控制探索的程度，确定置信水平。 如果$N_t(a)=0$，则a被认为是最大化动作。

上述这种做法的主要的出发点是平方根项是对不确定性的度量。可选动作a的真实的上限是最大化的数量，当每次做出动作选择a时，不确定性的值可能会降低，当选择该动作的次数上升时，不确定性项的值会降低。当选择动作a之外的其他动作时，

t的值会增大，但$N_t(a)$的值不变，不确定性的值会有所增大。随着时间的延续，自然对数的值会增大，增量会减小，但是却没有限制。直至最后将动作空间中的所有动作都采取了，那些具有较低估计值或者已经频繁选择的操作的选择频率会随着时间的推移而降低。

从文献[19]中可以看到，UCB 算法具有较好的效果。本文中将采用这种方法。

5、增量实现

按照贪心算法的做法，我们需要计算Q的值。在这里我们用Q_n表示在第 n 次动作选择的估计值，使用R_i表示在第i此选择该动作的获得的奖赏。如式(2-6)所示，

$$Q_n = \frac{R_1 + R_2 + \bullet\bullet\bullet + R_{n-1}}{n-1} \tag{2-6}$$

式(2-6)显示了我们需要记录的所有奖赏，但是当时间尺度增大时，对于计算和内存的要求就会上升。每个额外的奖励都需要开辟新的空间来存储。这种做法并不是必要的。可以采用以下新的计算方式达到目的。当给定Q_n和第 n 个奖赏R_n，所有 n 个奖赏的新的平均值可以由式(2-7)得到：

$$\begin{aligned} Q_{n+1} &= \frac{1}{n}\sum_{i=1}^{n} R_i \\ &= \frac{1}{n}(R_n + \sum_{i=1}^{n-1} R_i) \\ &= \frac{1}{n}(R_n + (n-1)\frac{1}{n-1}\sum_{i=1}^{n-1} R_i) \\ &= \frac{1}{n}(R_n + (n-1)Q_n) \\ &= \frac{(n-1)Q_n + R_n}{n} \end{aligned} \tag{2-7}$$

上式中只需要保存Q_n和n的存储，而只需要计算R_n的值就可以得到新的估计值。不管是 Softmax 算法还是$\varepsilon - greedy$算法，均采用这种增量实现方式来完成下一估计值的更新。

2.3 移动边缘缓存

2.3.1 移动边缘缓存发展背景

随着互联网以及通信技术的高速发展，人们对于在线视频等多媒体资源的需求越

来越高，根据相关预测，视频等高质量网络内容[39]流量将会达到新的顶峰。随着经济和制造工艺的发展，越来越多的人在使用智能设备进行上网，大规模的网络请求和网络流量对现阶段的网络架构和基础设施提出了新的挑战。在现阶段的通信服务方式中，用户在请求内容时多是经过基站向远端服务器发起请求，在大规模的用户请求下，会造成服务器负担过重，网络承受压力过大，服务器的处理速度满足不了用户的需求，进而导致用户请求的内容的时延增大，降低上网体验，甚至有可能导致服务器崩溃。如何让用户在短时间内快速得到请求的内容是目前网络和通信研究者的重要研究课题。另外，值得注意的是，在目前的网络形式中，社会热点、名人效应等热点问题会引起广大用户的极大兴趣并会促使他们对相关内容进行请求，即是大多数人访问较小一部分的内容。这样一来，网络中的流量充满了诸多的重复内容，为减少网络拥挤，提高用户体验，可以将大多数人请求的相同内容存储在特定设备上，在用户请求的时候就可以直接访问该存储设备，并不需要再次向远端服务器发出请求。这也就是移动边缘缓存技术的发展的重要原因，其具体模型如图 2.5。

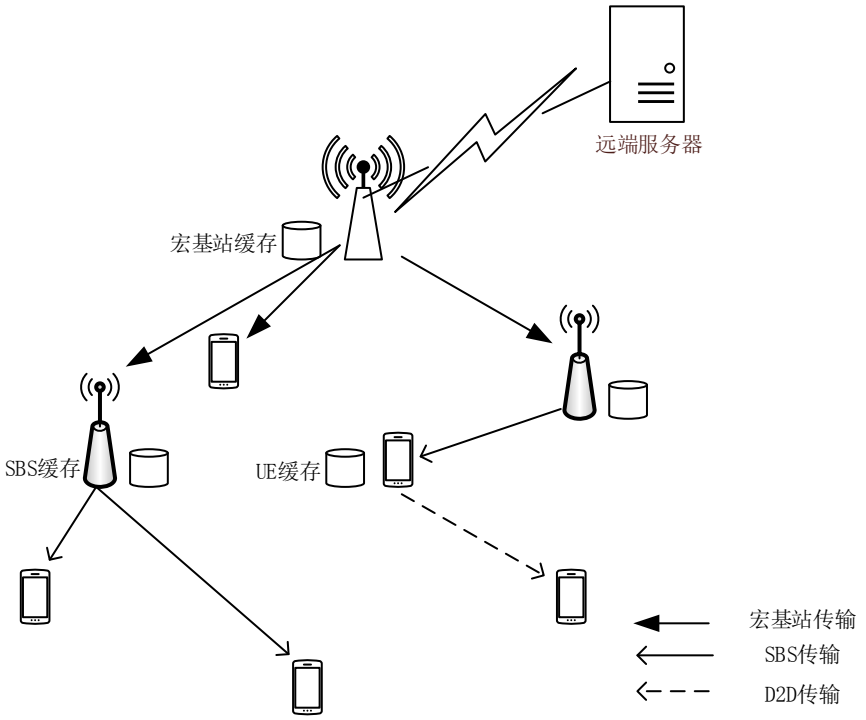

图 2.5　移动边缘缓存模型

2.3.2　移动边缘缓存的研究内容

从网络模型上看，移动边缘网络缓存主要包括宏基站缓存、小基站(SBS)缓存[40]和移动设备缓存。这三种缓存方式的共同目的就是存储内容以拉近用户和请求的内容之间的距离，减少网络拥挤，减轻服务器压力，提高内容的重复利用率，进而提高用

户体验。但是这三种缓存方式却各有特点，相互补充。宏基站由于功率大、辐射范围广，服务用户较多，相应的缓存设备存储空间也较大，接收的用户请求较多，这在一定程度上通过缓存尽可能多的内容增加了请求内容的命中率，减轻了服务器的压力。但这也使得宏基站缓存必须照顾所辐射范围内的所有用户，对基础设施能力要求较高。SBS 缓存的特点是距离用户更近，SBS 缓存的使用对整个蜂窝网有着重要的意义。通过部署众多 SBS 基站[41]可以使得用户更近的获得请求内容，减少用户向宏基站的请求。相应的在 SBS 基站处部署缓存设备可以更加有效的传输内容，对所覆盖区域内地用户负责。但是 SBS 基站由于基础设施比起宏基站相差甚多，存储设备空间有限，这限制了内容的存储数量，使得 SBS 基站在缓存时需要更加的注意所在区域用户的兴趣度，在有限的存储空间中缓存最有可能被请求的内容，充分利用自身的资源。移动设备缓存是最新发展的技术，它主要依托的是移动设备。随着科技的进步，智能设备的存储空间也愈来愈大，这使得一部分内容在用户有意愿的情况下可以进行适当地缓存，并且用户之间有更近的距离，若用户在请求内容时周边用户存储空间中有相应内容，在距离允许的情况下可以进行更快的 D2D 传输，这不仅提高了蜂窝网频谱利用率，而且使得用户体验也更好。但是 D2D 传输有着自身的缺点，它对自身的设备要求较高，而且用户是否愿意进行内容分享也是一大问题。

现阶段移动边缘网络缓存[42]主要研究的问题基本上分为三类：一是缓存位置的选取；二是缓存内容；三是缓存内容的更新。移动边缘缓存的目标就是尽快的给用户提供内容请求，这就使得缓存设备需要距离用户更近，现实中用户分布在不同的位置，不可能在每个位置上放置一个缓存设备，这就需要衡量好缓存设备与用户请求的关系，需要找到比较合适的位置来放置缓存设备[43]。在放置缓存设备的同时，如何有效的利用有限的缓存空间是需要重点考虑的问题，现阶段出现的解决办法有根据齐夫定律选取最流行的内容进行缓存等。另外，在缓存过内容之后，随着时间的推移，有些内容可能已经过时，被请求的概率已经相当小，这时候需要将其清除出存储空间以用来缓存较新的内容，但是需要研究的是如何认定已经缓存的内容是过时的以及什么时候将其替换。现阶段研究这使用较多的是 LRU 算法、LFU 算法以及 FIFO 算法等。在本文中，主要研究的问题是在移动设备缓存技术的基础上，研究缓存设备的选取问题和缓存内容的更新问题。

2.4 本章小结

本章主要介绍了使用到的相关技术。首先介绍了 D2D 通信技术的发展和应用场景，说明了现阶段发展 D2D 技术的重要意义，接着介绍了 D2D 通信技术中的设备发现和模式选择两种关键技术。然后介绍了强化学习的发展历史，并且详细介绍了强化

学习的模型以及其中模型元素的含义，之后介绍了几种强化学习算法。在本章的最后介绍了移动边缘网络缓存的发展背景和研究内容。通过对上述几种技术的详细介绍，给以后章节进行更深一步的研究打下了良好的基础。

第三章 基于亲密关系组的主从节点协作缓存模型

在现实社交关系中，具有亲密关系的人经常出现在彼此周围，例如，同学关系、邻里关系、同事关系等。绝大多数时间他们在同一个区域内学习、生活、工作，在这样的关系范围内建立有效的 D2D 通信方式，条件允许且可以有效缓解通信回程链路拥挤情况，减轻网络负担，提高资源利用率，丰富通信模式，一定程度上可以进一步加强通信双方的亲密关系。如何在亲密关系组内建立有效的通信缓存机制是本文关注的问题。本章主要关注如何选择缓存节点和在缓存节点中存储什么样的内容可以使整个关系组内传输的总延迟达到较低的水平。对于具有亲密关系的用户组来说，他们之间存在着一定的信任度，这一点从文献[5, 6, 7, 8]中可以看出，这些文献在考虑缓存节点的选取时均将亲密度或者信任度作为重要的条件。在存在信任的前提下，本文提出了基于亲密用户关系组的信息缓存机制。本章主要介绍系统模型、基于虚拟时延的最优缓存节点选择算法、主从节点协作缓存模型以及移动条件下的缓存节点选择策略。其中系统模型主要包括通信模型、缓存模型和奖赏模型。

3.1 系统模型

本小节主要介绍本文所用的模型。其中通信模型是建立 D2D 通信连接的基础，在这里主要考虑了用户的组成和建立通信的条件。缓存模型主要介绍了用户的请求分布和候选用户的组成。奖赏模型主要介绍了强化学习在缓存中的应用方式和缓存更新的细节。

3.1.1 通信模型

考虑这样一个用户组U，该组内含有 N 个用户$U=\{u_1,u_2,u_3,...,u_N\}$，每个用户之间存在着一定的亲密度。每一个用户既可以直接通过基站进行通信，也可以在周围亲密用户可以进行 D2D 通信的情况下建立 D2D 连接，进行内容的传输。用户组分为普通用户组和候选用户组。其中，普通用户组作为请求端，候选用户组作为响应端。具体模型如图 3.1。

我们假设用户u_m通过 D2D 通信连接的方式向用户u_n请求内容，则两者之间的距离$dis_{m,n}$必须满足条件$dis_{m,n}\le r$。r为两用户进行 D2D 通信能完整传输信息的最大距离。当$dis_{m,n}>r$时，我们认为用户u_m和用户u_n之间不能进行 D2D 通信，请求者只能通过基站请求感兴趣的内容。根据文献[43]，我们假设用户与基站之间没有干扰，因为 D2D 用户使用的频谱与 BS 站使用的频谱不同。本文将用户请求的内容做单位化

处理。

当用户 u_m 通过 D2D 连接向用户 u_n 请求内容时，连接传输速率可以表示为

$$R_{n,m}^{d2d} = W \log_2 (1 + \frac{P_n h_{n,m}^2 r_{n,m}^{-\alpha}}{\sum_{u_l \in U, n \neq l} P_l h_{l,m}^2 r_{l,m}^{-\alpha} + \sigma^2}) \tag{3-1}$$

其中 W 为传输链路的带宽，P_n 为用户 u_n 的传输功率。$h_{n,m}^2$ 为用户 u_n 与用户 u_m 之间的链路系数。$r_{n,m}^{-\alpha}$ 是用户 u_n 和用户 u_m 之间的路径损耗。$r_{n,m}$ 是用户 u_n 和用户 u_m 之间的物理距离。α 是路径损耗指数，σ^2 是每个用户的噪声功率。

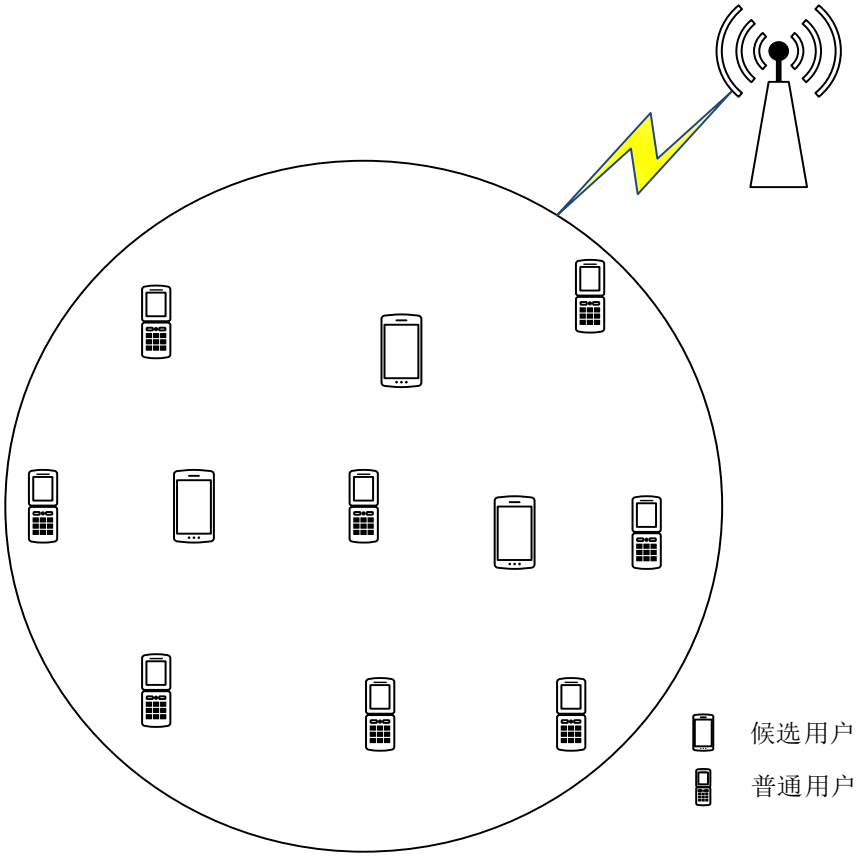

图 3.1　通信模型

用户 u_m 向用户 u_n 请求 p 个单位内容需要花费的时长为

$$d_{m,d2d} = p * del_{m,n} \tag{3-2}$$

其中

$$del_{m,n} = 1 / R_{n,m}^{d2d} \tag{3-3}$$

$del_{m,n}$ 为请求单位内容的延迟，p 为请求的单位文件总数。当用户 u_m 向 BS 站请求内容时，传输速率可以用一个固定值 $R_{0,m}^{\text{bs}}$ 表示。如果用户 u_m 向 BS 请求内容，则至少需要

$$d_{m,bs} = p * del_{m,bs} \tag{3-4}$$

其中

$$del_{m,\text{bs}} = 1/R_{0,m}^{bs} \tag{3-5}$$

我们忽略从基站到内容服务器的延迟。根据文献[11]，D2D 通信之间的时延比终端与基站之间对时延小的多，即 $d_{m,d2d}$ 远小于比 $d_{m,bs}$。

3.1.2 缓存模型

我们假设基站和一个内容服务器可以通信，内容服务器与互联网相连。假设内容服务器中有一个文件库 $F=\{f_1,f_2,f_3,...,f_K\}$，共有 K 个文件。文件分布满足齐夫定律。我们用 HU 表示候选用户集，a 表示意愿因子，a 取值为 0 或 1，代表用户在运营商的激励下是否愿意成为候选用户，则 $HU=a*U$。我们用 C 代表每一个候选用户的设备的存储能力 $C=\{c_{u_1},c_{u_2},c_{u_3},...,c_{u_t}\mid t<K,u_t\in HU\}$。为了便于表述，我们将内容服务器中的每一个文件做单位处理，对于所有用户的存储能力 $c_{u_i}<K$。当基站向缓存节点用户缓存文件 f_i 时，$f_i=1$ 表示文件已经缓存在设备中。对于用户群中每个用户 u_i，均有属于自己的一个兴趣偏好因子 γ_i，对应的文件请求概率 $p^{i,k}$ 满足齐夫定律：

$$p^{i,k}=\frac{1/k^{\gamma_i}}{\sum_{j=1}^{K}1/j^{\gamma_i}} \tag{3-6}$$

我们使用这样一个矩阵 $M\text{at}=\{p^{u_i,f_k}\}_{N\times F}$ 代表用户组中的各用户对文件库中的文件的请求概率。

3.1.3 奖赏模型

在提出的缓存机制中，主要对缓存替换起指导作用的是实时 MAB 模型。该算法在前面的章节中进行了简单介绍。实时 MAB 模型是一种基于环境反馈的强化学习模型。当一个智能体选择一个动作时，环境将根据动作的结果反馈给智能体。它将告诉智能体当前动作可以获得多少奖赏。智能体将根据当前结果进行下一步动作。随着训

练次数的增加，我们希望智能体能得到最大的收益。我们需要做出权衡，不仅充分利用现有资源，在拥有已有知识的情况下最大化收益，还要利用未知资源以期望在未来最大化收益。我们使用ε表示利用和探索的比率。对应在缓存问题中，缓存节点是智能体，动作空间是文件库中的文件是否缓存在缓存节点的缓存空间中。我们希望缓存节点做的缓存决策能给整个关系组带来最大的延迟降低量。在上一小节中提到每一个用户对文件库中的文件有自己的一个请求概率矩阵$\{p^{u_i,f_k}\}_{1\times F}$，且满足

$$\sum_{k=1}^{K} p^{u_i,f_k} = 1 \tag{3-7}$$

本文中使用L代表缓存节点的缓存文件集合。$I(t, p^{u_i,f_k})$作为指示函数表示用户u_i是否缓存了文件f_k。使用$D(t)$表示当前时隙下整个关系组中的用户所经历的整体时延。使用$D(u_i, t)$表示用户u_i在当前时隙下经历的时延。于是，有如下等式：

$$D(u_i, t)=\begin{cases} d_{i,d2d}, & I(t,p^{u_i,f_k})=1 \\ d_{i,bs}, & I(t,p^{u_i,f_k})=0 \end{cases} \tag{3-8}$$

对于整个关系组内的所有用户则满足下式，

$$D(t)=\sum_{i=1}^{N} D(u_i, t) \tag{3-9}$$

我们的目标是降低整体的请求时延，因此我们的优化目标为

$$\begin{aligned} &\min_{I} D(t) \\ &s.t.\ (a)\ \sum_{k=1}^{K} I(t, p^{u_i,f_k}) \le C_{re}, \quad i=1,2,3,...,N \\ &\quad (b)\sum_{k=1}^{K} p^{u_i,f_k} = 1, \qquad\qquad i=1,2,3,...,N \end{aligned} \tag{3-10}$$

其中，C_{re}代表缓存节点的存储容量。条件(a)表示缓存节点的存储容量不能超过其容量限制。我们希望每个时隙的整体延迟能够随着时间的推移而最小化。这要求我们必须有效地分配缓存文件来实现这个目标。根据以往的研究文献，仅仅在基站上缓存热点文件并不能取得很好的效果。即使用户偏好已知，上述的优化问题也是一个NP-complete 问题。但是，基于实时 MAB 的方法在优化问题上显示出了很大的优势。

先评估内容流行度再作缓存决策的方法使得缓存决策并不是最优解，且收敛速度较慢。因此，我们希望在缓存空间中根据环境的反馈直接做出缓存决策，而不需要事先估计用户的偏好。我们把环境的反馈看作是一种奖赏。这样，当缓存节点缓存文件时，我们使用延迟减少量作为系统的奖赏：

$$Reward(t) = p*\left[\sum_{i=1}^{N}\sum_{k=1}^{F} I(t, p^{u_i, f_k})(del_{i, bs} - del_{i, d2d})\right] \tag{3-11}$$

在获得奖赏后，我们会将缓存文件带来的奖赏进行迭代保存，从而统计相应文件给系统带来的总奖赏值。当我们决定缓存或更新缓存文件时，我们可以以此作为参考。本文认为，如果相应文件的奖赏值很高，那么缓存该文件将会给整体带来更大的奖赏。这说明关系组内的用户对该文件有更多的偏好。因此我们的目标从上述的优化问题转变为求取最大奖赏的问题：

$$\begin{aligned} &\max_{I} Reward(t) \\ &s.t.\ (a)\ \sum_{k=1}^{K} I(t, p^{u_i, f_k}) \le C_{re}, \quad i = 1,2,3,...,N \\ &\quad (b)\ \sum_{k=1}^{K} p^{u_i, f_k} = 1, \qquad\qquad i = 1,2,3,...,N \end{aligned} \tag{3-12}$$

在这里我们采用强化学习中的试错迭代思想，通过足够多的训练次数来逐渐接近最优解。下面介绍使用实时 MAB 算法中来实现的缓存更新原则。

本文希望在未知内容流行度的情况下直接学习缓存策略，经过一段时间的训练，缓存空间中所存储的内容最符合当前关系组整体利益。在这里，本文将详细给出缓存替换细节。

在每一个时隙内，关系组内的用户均有可能发起请求，我们要做的是通过决定缓存内容来提高文件命中率。因此，在决定缓存时我们希望存储的是能给我们带来最大奖赏的内容。这里，我们在做缓存决策时采用了 $\varepsilon - greedy$ 算法。当用户发起请求，通过计算该请求可以带来的奖赏值来决定是否存储或更新。在更新每一次的奖赏值时，采用了第二章中强化学习小节中提到的增量更新。并且根据文献[5]，在某种程度上两用户之间的距离代表了用户之间的亲密度。具体如式(3-13)所示：

$$In(r) = \begin{cases} D^2/r^2, & r > D \\ 1, & 0 < r \le D \end{cases} \tag{3-13}$$

其中， $In(r)$ 代表两用户之间的亲密程度。 D 为预定义的距离，它表示请求者和缓存节点具有的稳定社会关系。基于式(3-13)，本文采取对用户请求获得的奖赏值进行加权处理，这样做不仅考虑了用户的请求，而且还考虑到用户的社会关系，使得奖赏值对于缓存决策更具参考价值。具体表示如式(3-14)：

$$V(I(t,p^{u_i,f_k}))=\frac{(t-1)*V+Reward(t)*In(r)}{t} \tag{3-14}$$

具体缓存更新细节如下：

基于实时MAB的缓存更新算法

```
输入：
    用户亲密关系组、文件库
初始化：
    中心节点的缓存空间、奖赏空间、动作空间
for t=1,2,3,…,T do
    根据ε-greedy算法选择一个文件f
    计算此时的奖赏
    for 每一普通用户 do
        用户根据自己的兴趣偏好矩阵请求文件
        计算时延
    根据奖赏公式(3-14)更新奖赏
    if 缓存空间未达到上限 then
        if 文件f已被缓存 then
           continue
        else
           将文件f缓存
    else
        决定文件f是否被缓存
        if 文件f已被缓存 then
            continue
        else
            找到具有最小的奖赏值的已缓存文件fmin
            if 文件f的估计奖赏值大于fmin的奖赏值
                使用文件f替换文件fmin
    记录每一时隙请求时延与奖赏；
    维护基站处各列表；
返回中心节点缓存决策
```

图 3.2　基于实时 MAB 的缓存更新

图 3.2 中显示的算法主要特点是不需要评估内容流行度，而是直接根据用户的实际请求进行缓存更新。该算法节省了评估内容流行度的资源，也较好的适应了现实生活中人们对兴趣的动态性和突然出现的社会热点。该缓存更新算法在主从缓存模型中进行了应用，并且取得了良好的效果，具体效果可见第四章的仿真结果。

3.2 基于实时 MAB 的主从节点协作缓存模型

本节主要介绍本文所提出的两个算法，基于虚拟时延的最优缓存节点选择算法和主从节点协作缓存算法。节点选择算法采用了虚拟时延的思路，根据缓存更新策略计算候选集中用户的缓存服务能力，然后将候选集中的用户进行排序。根据排序结果，确定主节点，接着进行主从节点的协作缓存以进一步提高整体的缓存命中率。

3.2.1 缓存节点的选择策略

候选集的用户在运营商的激励下，愿意成为缓存节点。但是，从用户关系组和操作成本的角度来看，并不适合每个候选用户成为缓存节点。从以前的文献可知，从候选用户中选择一个或多个缓存节点是合适的。根据文献[5]，根据用户的信任值和设备能力选择缓存节点。被选中的用户需要放弃他们的兴趣点。文献[6]对于缓存节点的选择也有类似的方法，它考虑了用户之间的信任值、兴趣相似度和设备能力。但是，缓存节点的选择方法需要大量的用户历史记录。而且，历史记录并不能准确地解释用户的偏好。随着时间的推移，一些用户可能会随着年龄的增长而改变自己的兴趣。通过上述方法选择的缓存节点可能不是整个组的最优选择。另外用户自身也存在着一定的自私性，比如设备的存储能力和隐私安全等因素，也都制约着普通用户自愿成为缓存节点。因此，对于候选集中的用户也并不一定都会成为缓存节点，我们需要做出一定的筛选来决定缓存节点，也为后期的多节点协作缓存奠定基础。针对上面的问题，本文提出了一种基于虚拟时延的最优缓存节点选择算法。

所谓虚拟时延指的是候选用户集中只有一个用户与请求者建立实际 D2D 连接，在本文中为了评估候选用户的缓存服务能力，假设其他候选用户也与请求者建立了 D2D 连接，从而得出的虚拟时延和奖赏。因为在同一个用户关系组内由于距离相隔不会太远，信道环境基本相同，虚拟连接的传输速率可以参考实际 D2D 连接。我们假设有亲密关系的用户之间存在信任，而每个用户不必放弃自己的偏好。为了真实反映当前用户的兴趣，我们选择对候选集合中的每个用户进行实时评估，评估他们作为单个缓存节点对整个关系组的缓存服务能力，并在基站收集他们的评估结果。然后根据评估结果确定缓存节点。为了测试候选用户对整个关系组的缓存服务能力，首先进行第一阶段的评估测试。在这里所要进行的操作是：对整个关系组来说，候选集像是一个整体缓存节点，但在候选集内部将其视为在不同位置的分布式缓存。简要流程如图 3.3。基站所做工作如图 3.4。基站处需要维护的信息结构如图 3.5 所示。

在基站处为候选集中的每个用户维护一个列表，用于表示该用户缓存的内容。当普通用户发起一个请求时，基站在检测到请求后，会查看在该用户周围的候选用户中是否含有该内容，若存在则按照距离优先的原则在基站的控制下建立 D2D 连接进行

数据传输。若候选用户中并不包含请求的内容，则从基站处请求内容，基站处则相应维护与其最近的候选用户列表中请求的内容，在非高峰期将内容传输到候选用户的缓存中。若是候选集终端用户发起请求，则首先检测本地缓存是否含有请求内容，若没有则向基站请求。同样需要在非高峰期将请求内容缓存在本地。在每一时隙记录关系组内每一个用户的请求时延。

基站除了为每一个候选用户维护列表外，也为在整个基站覆盖范围内的各个亲密关系组维护一个总的请求列表，以记录请求文件和请求用户的之间的关系，此记录可以为用来分析各个关系组的内容流行度和指导各节点缓存。基站还需计算每个候选用户可能获得的总时延，以此来进行候选节点排序。具体处理关系为：当一个用户发出请求时，若它的范围内有多个候选用户，若请求未命中，则从基站获取内容。相应的，基站在非高峰期将向候选集中所有缓存设备中发送相同的内容进行缓存；若请求命中，在多个候选用户的条件下，根据距离最近优先原则建立 D2D 连接。但对于在候选集中的其他用户，同样会进行虚拟时延的计算，即对于其他的候选用户，基站衡量该候选用户若与请求用户之间建立 D2D 连接所带来的虚拟时延和奖赏。这里需要注意的是每一个候选用户的存储能力不同，在基站向候选集中的缓存节点传输缓存内容时，有的会达到其存储上限，这个时候，按照实时 MAB 算法进行相应的缓存替换即可。在训练一段时间后，候选集中的用户对于整个关系组的缓存服务能力的大小就会得到确定。

从上述过程中我们得到了节点的缓存服务能力的列表，该列表的获得是根据实际的用户请求来决定缓存的内容，在一段时间的训练后得到了各节点的缓存服务能力。这更加符合实际要求，能很好的把握当下周围用户的兴趣点，具有较好的时效性。基于虚拟时延的最优缓存节点选择算法比文献[5,6]中的节点选择方法的先进之处主要有三点。第一，本文提出的节点选择算法是以整个用户亲密组为对象，侧重整体延迟的降低。实际生活中若一个系统可以使得整体利益最大化，从运营商的角度来看会更容易推行。第二，本文提出的节点选择算法是以用户的实际请求来考察候选用户的缓存服务能力。在算法执行执行过程中并不依赖于历史记录，而是跟随用户的兴趣和社会热点等因素的变化而动态调整缓存内容，进而得出候选用户的缓存服务能力排序列表。第三，本文提出的节点选择算法并没有首先评估内容的流行度而是根据实际请求的反馈来决定缓存内容。本文将在第四章的仿真分析中来验证所提算法的有效性。

3.2.2 主从节点协作缓存模型

从文献[11,12,14]可知，缓存节点的选择并不是唯一的。它允许多个缓存节点同时存在。并且从文献[41]中可知，在多个缓存节点存在的情况下，若各节点单独进行缓存，并不考虑其他缓存节点的的已经存储的内容，这时各节点之间的内容的多样性会

受到限制，会存在大量的内容重复，这在存储空间有限的条件下明显降低了空间利用率，进而会影响整个关系组的性能的提高。因此考虑多节点协作缓存是很有必要的。在亲密关系组的场景下，第一阶段的评估测试计算了各候选节点的缓存服务能力。缓

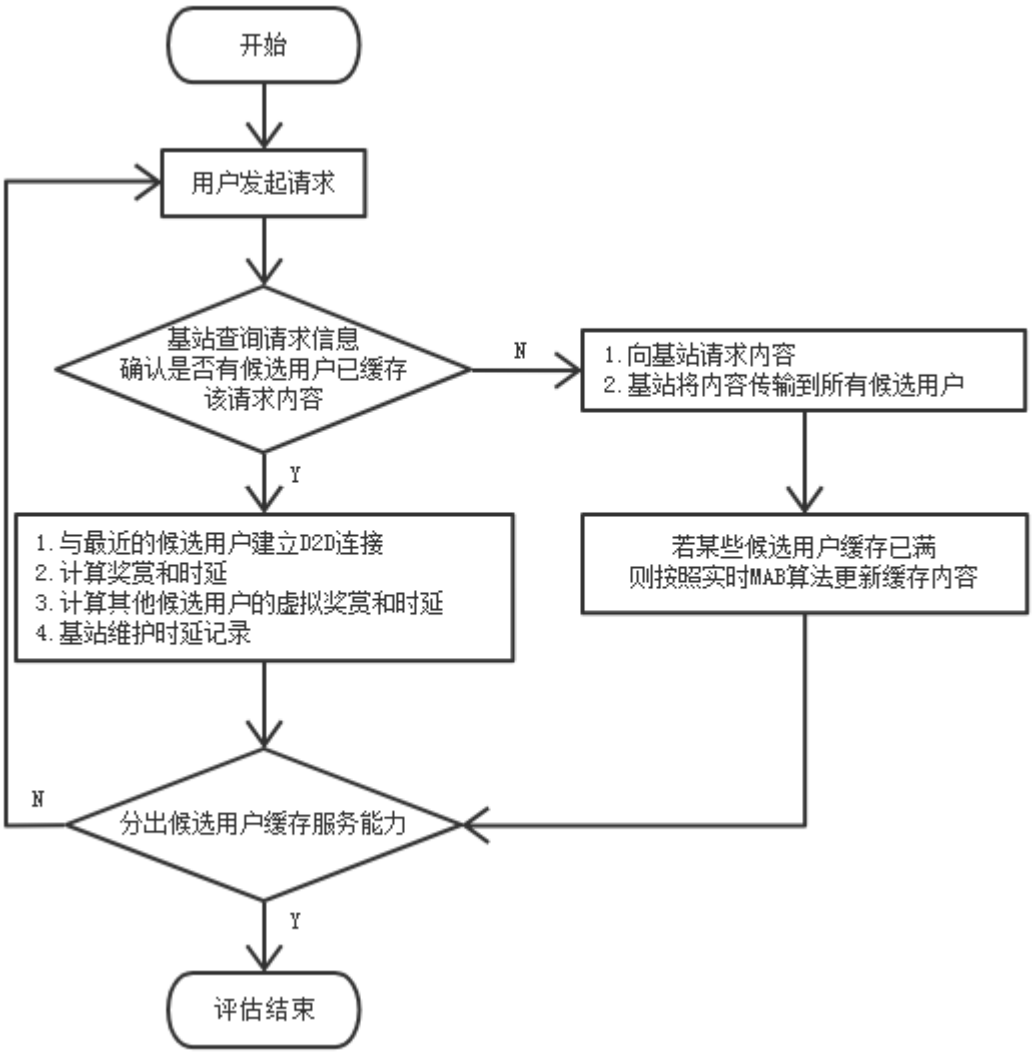

图 3.3 节点评估流程

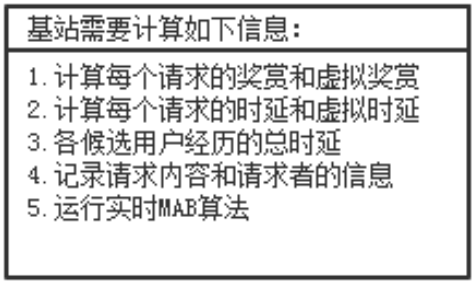

图 3.4 基站工作信息

存服务能力越强，说明该节点对整体越重要。因此，在进行多节点协作缓存时，有必要对贡献能力的强的节点重点利用。因此，在本小节提出了一种主从节点协作缓存模型，如图 3.6。

因为整个用户组的用户彼此相距不远，所以我们假设三个缓存节点可以覆盖整个

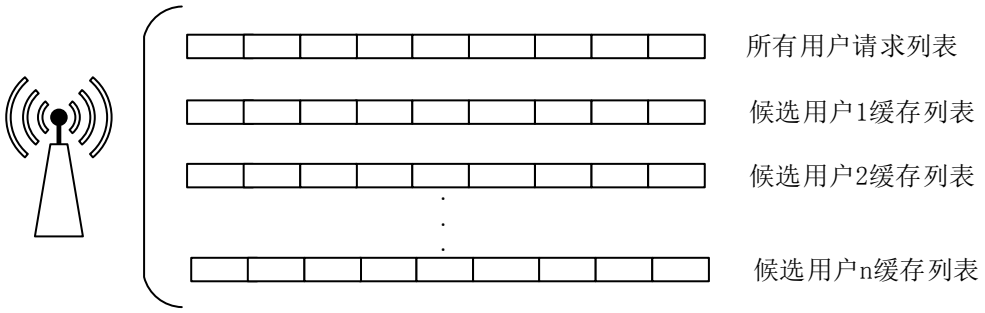

图 3.5　基站维护的信息

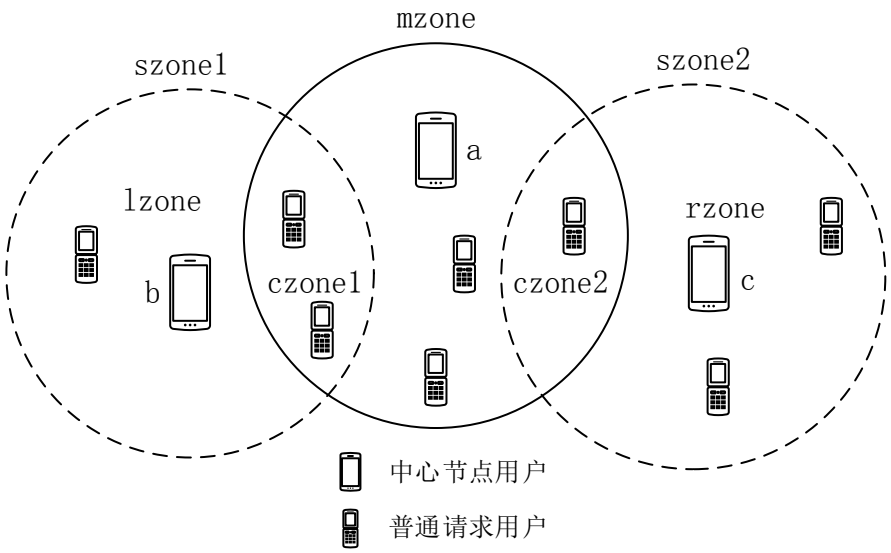

图 3.6　主从节点协作缓存模型

用户组。在该模型中，我们定义缓存节点用户 a 为主节点，缓存节点用户 b 和 c 为从节点。a 节点所覆盖区域为 mzone，b 节点和 c 节点所覆盖区域分别为 szone1 和 szone2。mzone 区域和 szone1 区域、szone2 区域的交叉区域分别为 czone1 区域和 czone2 区域。我们定义 lzone 区域为 szone1 区域除去 czone1 区域范围，rzone 区域为 szone2 区域除去 czone2 区域范围。

该模型主要针对缓存内容决策阶段，主从节点的选择依据上一小节中的缓存服务能力排序列表。在这里，我们将缓存服务能力最强的节点作为主节点，从节点的选择依据在缓存服务能力差距不大的情况下尽量选择 D2D 通信范围覆盖用户多的节点，否则按照顺序排序进行选择。

在多节点协作时，缓存的更新算法按照实时 MAB 算法进行。在第一阶段评估的基础上，我们已经得到各缓存节点的内部存储的内容，接下来需要进行的操作是将主节点和从节点确定下来，在模型中，确定缓存节点 a 为主节点，缓存节点 b 和 c 为从节点。从候选用户中选取出来的节点均称作缓存节点。

主从节点协作缓存算法

```
输入：
    候选用户排序列表
    排名前三候选用户缓存内容
初始化：确定主节点和从节点
        删除从节点中与主节点相同的缓存内容;
for t=1,2,3,…,T do
    for 每一普通用户发出请求 do
        if 该用户周围有中心节点 then
            if 中心节点包含主节点 then
                if 请求在主节点命中 then
                    建立D2D连接
                else if 请求从节点命中 then
                    按照距离最近原则建立D2D连接
                else
                    向基站请求
                    基站按缓存更新算法决定是否向主节点更新缓存
                    if 主节点没达到缓存更新要求 then
                        基站按缓存更新算法决定是否向从主节点缓存
            else
                if 从节点命中 then
                    建立D2D连接
                else
                    按照距离最近原则与从节点建立D2D连接
        else
            向基站请求
    记录每一时隙个请求时延与奖赏;
    维护基站处各列表;
返回总体时延和主从节点缓存决策
```

图 3.7 主从节点协作缓存算法

以主节点内容为基准，删除从节点中与主节点中一样的内容，这是第一步。接着，当普通用户进行请求时，若该用户可以进行 D2D 通信范围内有多个缓存节点，此时，需要分为两种情况，一种是包含主节点，另一种是不包含主节点。在包含主节点的情况下，也就是处在 czone1 区域和 czone2 区域的用户，用户优先与主节点建立 D2D 连接，若主从节中均未缓存该项内容，则由基站处记录值按照奖赏替换原则来决定是否进行缓存更新。若达不到更新的情况，则在距离最近的从节点 b 或 c 中缓存该内容，其更新模式与主节点 a 一样。在不包含主节点 a 的情况下，即处在 lzone 区域或者

rzone 区域中的用户，在从节点 b 或者 c 均未命中时，请求按照距离优先原则进行缓存与替换。特殊情况在于，lzone 区域和 rzone 区域的用户请求的内容已经在主节点 a 存储，但是却不在主节点 a 的覆盖范围内，此时应按照缓存替换原则进行操作，若该请求的内容在从节点处也到了很高的估计奖赏值则应该在从节点缓存该内容。此小节提到的缓存替换原则核心思想是在当前缓存节点的存储容量达到上限时，若该请求内容的估计奖赏值超过了当前存储设备中的存储的内容的最低奖赏值，则将该内容与具有最低奖赏值的内容进行替换。具体算法如图 3.7。接着，本文将根据主从节点协作缓存算法的核心思想建立奖赏模型。

根据图 3.6 所示，用户主要分为三部分，主区域特有用户、交叉区域用户、从节点剩余区域用户。主区域内的用户在请求时主要有两种请求方式：向主节点请求和向基站请求。在这里定义为

$$MU=\{u_i \in mzone,\ u_i \notin czone1,\ u_i \notin czone2 \mid u_i \in U\} \tag{3-15}$$

其对应的请求延迟为

$$D_{MU}(t)=\sum_{u_i \in MU} D_{MU}(u_i,\ t) \tag{3-16}$$

其中，

$$D_{MU}(u_i,\ t)=\begin{cases} d_{i,\mathrm{a}}\ , & I(t,p^{u_i,f_k})=1 \\ d_{i,bs}, & I(t,p^{u_i,f_k})=0 \end{cases} \tag{3-17}$$

交叉区域内的用户在请求时主要有三种请求方式：向主节点 a 请求、向从节点 b 或 c 请求、向基站请求。我们将此区域内的用户定义为

$$CU=\{u_i \in czone1, u_i \in czone2 \mid u_i \in U\} \tag{3-18}$$

向主节点 a 请求带来的时延为

$$D_{CU}(t)=\sum_{u_i \in CU} D_{CU}(u_i,\ t) \tag{3-19}$$

其中

$$D_{CU}(u_i,\ t)=\begin{cases} d_{i,\ a}\ ,\ I_a(t,p^{u_i,f_k})=1 \\ d_{i,\ b}\ ,\ I_b(t,p^{u_i,f_k})=1 \\ d_{i,\ c}\ ,\ I_c(t,p^{u_i,f_k})=1 \\ d_{i,\ bs},\quad I(t,p^{u_i,f_k})=0 \end{cases} \tag{3-20}$$

$I_a(t,p^{u_i,f_k})=1$ 代表请求的文件 f_k 存储在主节点 a 中。$I_b(t,p^{u_i,f_k})=$ 和 $I_c(t,p^{u_i,f_k})=1$ 均代表类似的含义。$I(t,p^{u_i,f_k})=0$ 表示请求的文件 f_k 在主从节点均没有被缓存。

从节点剩余区域内的用户在请求时主要有两种请求方式：向从节点 b 或 c 请求和向基站请求。我们将此区域内的用户定义为

$$RU=\{u_i\in lzone,\ u_i\in rzone \mid u_i\in U\} \tag{3-21}$$

其对应的请求延迟为

$$D_{RU}(t)=\sum_{u_i\in RU} D_{RU}(u_i,\ t) \tag{3-22}$$

其中

$$D_{RU}(u_i,\ t)=\begin{cases} d_{i,\ b}\ 或 d_{i,\ c}\ ,\quad I_b(t,p^{u_i,f_k})=1 或 I_c(t,p^{u_i,f_k})=1 \\ d_{i,\ bs},\qquad\qquad I(t,p^{u_i,f_k})=0 \end{cases} \tag{3-23}$$

从以上分析可以得出我们的优化目标为

$$\begin{aligned} &\min_I D(t)=\min_I[D_{MU}(t)+D_{CU}(t)+D_{RU}(t)] \\ &s.t.\ (a)\ \sum_{k=1}^{K} I_a(t,p^{u_i,f_k})\ \le\ C_a,\quad i=1,2,3,...,N \\ &\quad (b)\ \sum_{k=1}^{K} I_b(t,p^{u_i,f_k})\ \le\ C_b,\quad i=1,2,3,...,N \\ &\quad (c)\ \sum_{k=1}^{K} I_c(t,p^{u_i,f_k})\ \le\ C_c,\quad i=1,2,3,...,N \\ &\quad (d)\ \sum_{k=1}^{K} p^{u_i,f_k}=1,\qquad\qquad i=1,2,3,...,N \end{aligned} \tag{3-24}$$

C_a、C_b和C_c分别代表主节点 a、从节点 b、从节点 c 的存储容量。

在解决上述优化问题时同样采取了和式(3-10)相同的策略，将最小化时延目标转化为最大化奖赏问题，然后通过实时 MAB 算法进行训练迭代以期得到缓存决策结果。本节所提的主从节点协作缓存算法比已有的协作缓存算法的优势主要有三点。第一，本文所提的协作缓存算法着重考虑了缓存节点自身的缓存服务能力的差异，充分发挥了各缓存节点的能力。第二，本文所提的协作缓存算法是多节点协作缓存。在用户关系组内若是多个节点同时存在，即可根据缓存服务能力的大小进行主从协作缓存，可以覆盖到更多的用户。第三，主节点在缓存决策时利用了节点选择时的缓存内容，节省了重新决策的资源，更快的掌握了周围用户当前的兴趣分布。

3.3 移动条件对缓存节点的选择和缓存决策的影响

从上面的分析中，我们得出候选用户周围的普通用户的兴趣偏好、请求概率、候选用户的自身设备条件以及用户之间的亲密关系等因素对缓存节点选择和缓存决策产生了重大的影响。但是在以上场景中均是默认用户是静止不动的，而实际场景中用户是具有移动性的，因此有必要对移动场景下的缓存节点选择和缓存决策作进一步的研究。

3.3.1 移动背景

现实中用户具备移动性，在亲密用户关系组内移动者分为普通用户的移动和候选用户的移动。由于我们研究的是普通用户和候选用户之间的通信连接，当其中一方进行移动时，就会影响另一方是否还能与其建立起通信连接，且其自身与基站之间的绝对速度也会对能否建立起与基站的通信连接有影响，所以本文采用绝对运动和相对运动模型来描述他们之间的移动变化。将用户的移动场景分为高速、中速和低速场景。本文采用v_1表示用户与基站建立起稳定通信连接的运动速度上限，v_2表示用户之间建立起稳定 D2D 连接的相对速度上限。当用户与基站的绝对速度超过v_1时，本文认为用户此时处于高速移动场景下，比如用户在坐动车或高铁等超快交通工具，用户并不能从基站处获得请求内容。当用户与基站的绝对移动速度在v_1之下，此时用户若乘坐一般的交通工具，如自行车，公交车等，用户处于中速移动场景下，这种场景下若建立 D2D 通信对用户之间的相对运动速度要求较高。在本文提出的亲密用户关系组内，当亲密用户之间的相对运动速度之间不超过v_2，且满足距离上满足通信条件，此时用户可以建立 D2D 通信交换信息。但是当 D2D 用户与基站间的绝对运动速度超过v_1时，在请求未命中时只能放弃通信。高速场景下只在特定情况下才出现且对于亲密用户关系组来说这是一种特例，因此本文主要研究中低速场景下，用户的移动性对缓存

节点的选择和缓存决策的影响。用户移动模型如图 3.8。

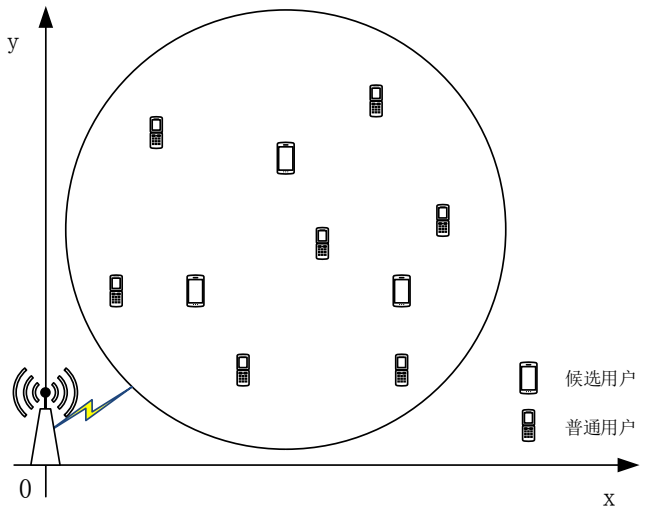

图 3.8 用户移动模型

本文采用随机游动模型来描述用户的移动行为。该模型主要是用户在当前位置上在特定速度范围内随机选择一个速度，并随机选择一个方向进行移动，每次移动可选特定的移动时间。以基站为中心，建立笛卡尔坐标系，如图 3.8 所示。假设用户集中的某一用户 u_m 在时隙 t 的起始位置为 (x_m^t, y_m^t)，相对移动速度为 v_m^t，方向角度为 θ，每次移动的时间为 T。则下一时隙用户所处的位置为 (x_m^{t+1}, y_m^{t+1})。表达式如式(3-25)和(3-26)。

$$x_m^{t+1} = x_m^t + v_m^t \bullet T \bullet \cos\theta \tag{3-25}$$

$$y_m^{t+1} = y_m^t + v_m^t \bullet T \bullet \sin\theta \tag{3-26}$$

假设候选用户集中的用户坐标为 (x_h^t, y_h^t)，则它与候选用户之间的距离为

$$r_{m,h}^t = \sqrt{\left(x_h^t - x_m^t\right)^2 + \left(y_h^t - y_m^t\right)^2} \tag{3-27}$$

则根据静止模式下的通信模型 $del_{m,d2d} = 1/R_{n,m}^{d2d}$，可以得到移动模式下用户之间的时延为

$$de_{m,h}^t = p / R_{n,m}^{d2d} \tag{3-28}$$

其中 p 为用户请求的内容数。若用户的请求未命中，则需要向基站请求内容，则与基站之间的时延为

$$de_{m,bs}^{t} = p / R_{0,m}^{bs} \tag{3-29}$$

其中

$$r_{m,bs}^{t} = \sqrt{\left(x_m^t\right)^2 + \left(y_m^t\right)^2} \tag{3-30}$$

当普通用户 u_m 与候选用户 u_h 之间的相对运动速度超过 v_2 时，此时两者之间并建立不了 D2D 通信连接，此时用户只能通过基站进行内容请求。综上所述，可以得出在移动模式下在时隙为 t 时的用户的请求时延为

$$D(u_{i'}\ t)=\begin{cases} de_{i,\ d2d}, & v_i^t < v_2,\ I(t, p^{u_i, f_k}) = 1, \quad i = 1,2,3,...,N \\ de_{i,\ bs}, & v_i^t > v_2, \qquad\qquad i = 1,2,3,...,N \\ de_{i,\ bs}, & I(t, p^{u_i, f_k}) = 0, \qquad i = 1,2,3,...,N \end{cases} \tag{3-31}$$

当用户 u_m 向候选用户 u_h 请求时，即 $i = m$ 时，$de_{i,d2d} = de_{m,h}$；同理，当用户 u_m 向基站请求时 $de_{i,bs} = d_{m,bs}$。本文使用 I' 表示 v_i^t 是否大于 v_2，如式(3-32)。

$$I' = \begin{cases} 1, & v_i^t \le v_2 \\ 0, & v_i^t > v_2 \end{cases} \tag{3-32}$$

于是，对于整个关系组内的所有用户则满足式(3-33)，

$$\begin{aligned} D(t) &= \sum_{i=1}^{N} D(u_{i'}\ t) \\ &= \sum_{i=1}^{N} \left[I(t, p^{u_i, f_k}) \bullet I' \bullet de_{i,\ d2d} + de_{i,\ bs} \right] \end{aligned} \tag{3-33}$$

本文的目标是使得整个亲密关系组的整体时延达到最小，所以优化目标为

$$\begin{aligned}&\min_{I} D(t)=\min_{I}\sum_{i=1}^{N}\left[I(t,p^{u_i,f_k})\bullet I^{'}\bullet de_{i,d2d}+de_{i,bs}\right]\\&s.t.\ (a)\ \sum_{k=1}^{K}I(t,p^{u_i,f_k})\ \le\ C_{re},\quad i=1,2,3,...,N\\&\quad\ (b)\ \sum_{k=1}^{K}p^{u_i,f_k}=1,\qquad\qquad i=1,2,3,...,N\end{aligned}\tag{3-34}$$

根据上一小节对分析，该问题将转化为奖赏模型，即为求得上述模型中的解将其转换为求得最大奖赏时的解。在奖赏模型下可以采用实时 MAB 算法进行求解。

3.3.2 缓存更新

从上一小节的分析，已经得出了整个关系组的时延模型，为了使得时延达到最小，我们需要缓存合适的内容使得用户的命中率尽可能的大，但是在移动模式下，用户的请求会随着位置的变化导致用户建立 D2D 连接的另一端用户发生变化，且在相应设备中缓存的内容也随之发生改变。静止模式下的缓存更新算法并不完全适用在移动模式下，因此接下来本文对缓存更新算法进行了一定的更新，使其适用于用户移动模式场景。

移动模式下主要涉及到的问题一个是用户的位置和速度的变化，另一个就是在建立连接后传输内容的变化。第一个问题已经在上一小节给出了解决办法。在两用户建立 D2D 通信连接后，由于用户的位置移动、运动速度发生了变化，有可能在还没接收完内容时用户之间的连接突然断开或者在还没有建立连接时就已经离开了缓存节点的范围，这时它需要在进入另一缓存节点的范围后再次进行 D2D 连接请求并传输后续内容。在相应的缓存节点选择策略中，我们在计算虚拟时延时，若在当前时隙用户并不能简单的将其视为本节点缓存的内容。本文为便于描述，将用户请求的文件进行了单位化，用户请求的文件可能有多个，本文假设每一次建立 D2D 连接最小传输一个单位文件。本文采取的更新措施为：若当前用户在内容传输阶段离开了当前缓存节点的范围，则当前缓存节点对该请求内容并没有奖赏。本文使用 f 表示用户是否在请求多个内容时离开了当前缓存节点。 $f=1$ 代表当前时隙下用户还没接收完完整内容就离开了当前缓存节点。反之，$f=0$。本文使用 j 表示用户请求的内容中已经接收的内容。其中， $j\le p$。则相应的奖赏模型更新如式(3-35)。

$$Reward(t)=\frac{j}{p}*\left[\sum_{i=1}^{N}\sum_{k=1}^{F}I(t,p^{u_i,f_k})(de_{i,bs}-de_{i,d2d})\right]\tag{3-35}$$

根据我们的奖赏优化模型和实时 MAB 算法得出缓存决策。在第四章的仿真分析中将证明本文所提的优化模型的有效性。

3.4 本章小结

本章主要针对现阶段缓存研究方面出现的问题提出来自己的解决方案，并进行了详细的理论推导和仿真验证。本章一开始先建立了通信模型、缓存模型、奖赏模型和缓存更新算法。在这几种基础模型的基础上，接下来提出了基于虚拟时延的最优缓存节点选择算法，并给出了详细的分析过程。在此基础上，根据所得的缓存节点的排序列表，提出了主从节点协作缓存模型，该模型主要考虑充分利用主节点的贡献能力，并结合从节点进行协作缓存以提高协作缓存服务能力，在该过程中列出了详细流程和算法细节。最后更深一步研究了移动条件下用户时延优化模型和奖赏模型的改进问题，用户移动的不确定性给缓存节点的选择带来了挑战，并进而影响了缓存内容的决策，在分析该问题时给出了详细的优化方法。下一章计划对本章内容进行仿真验证。

第四章　仿真结果及分析

本章主要对第三章提出的算法和模型进行仿真验证并进行一定的分析。主要内容包括参数的配置、对比方案以及仿真结果分析。

4.1 参数配置

本小节主要对第三章提出的实时 MAB 算法、缓存节点选择算法和主从节点协作缓存模型进行验证。仿真试验区域大小为$100\times100\text{m}^2$，在该区域内主要有一个亲密关系组，在组内有 10 个用户，分布在半径为 30m 的区域内。用户的缓存空间集合为$storage=[10, 16, 12, 16, 13, 11, 17, 14, 9, 15]$，每个数值代表可以缓存的文件数，缓存的文件大小假设为单位 1。其中候选用户集合为$candidates=[0,3,4,6,8]$，其中数字代表缓存空间集合的索引，代表某个用户。候选用户集合的索引使用数字编号(1,2,3,4,5)来表示。用户的兴趣偏好满足齐夫定律，齐夫参数选择为(0.3, 0.5, 0.7)。我们假设在文件库中有 50 个单位文件。每一个候选用户缓存的文件最大不能超过 20 个单位。这里将进行 D2D 通信的最大距离设置为 30m。具体如表 4.1。

表 4.1　系统参数

参数	值
用户组内的用户数 N	10
D2D 最大传输距离 r	$30m$
传输功率 P_n	$2W$
路径损失指数 α	4
带宽	$10^5\,Hz$
噪声系数 σ^2	10^{-10}

本文的使用的仿真环境为 Spyder 3:3:2，运行机器配置为 Intel(R) Core(TM) i5-4460 CPU @ 3.20GHz with RAM of 8GB。

4.2 对比方案

本文主要采用的对比方案为 LRU、LFU、缓存节点随机选择策略、多点独立缓存策略。LRU 和 LFU 主要用来验证实时 MAB 算法的实用性。缓存节点随机选择策略和多点独立缓存策略主要对本文提出的缓存节点选择策略和主从节点缓存策略进行效果对比。LRU 算法的核心思想是若数据在最近一段时间内没有被访问，那么在将

来也不会被访问；若是在近期被访问过，说明该内容的优先度较高。LFU 算法的核心思想是若数据最近一段时间内被访问的次数很少，那么在将来被访问的可能性也很小。缓存节点随机选择策略主要的策略是只要用户愿意成为缓存节点就将其作为缓存节点加入到多点协作策略中。多点独立缓存策略主要的想法是各缓存节点独立缓存，相互缓存的内容之间并不影响。

4.3 仿真结果

本小节主要对静止条件下的用户关系组进行仿真验证。主要比较了候选节点的缓存服务能力，为做出节点选择打下基础。在此基础上比较了各候选节点协作缓存的能力大小。除此之外还仿真了与各对比方案的比较实验以及不同齐夫参数和不同探索比例下对整体时延的影响实验。具体仿真结果如下：

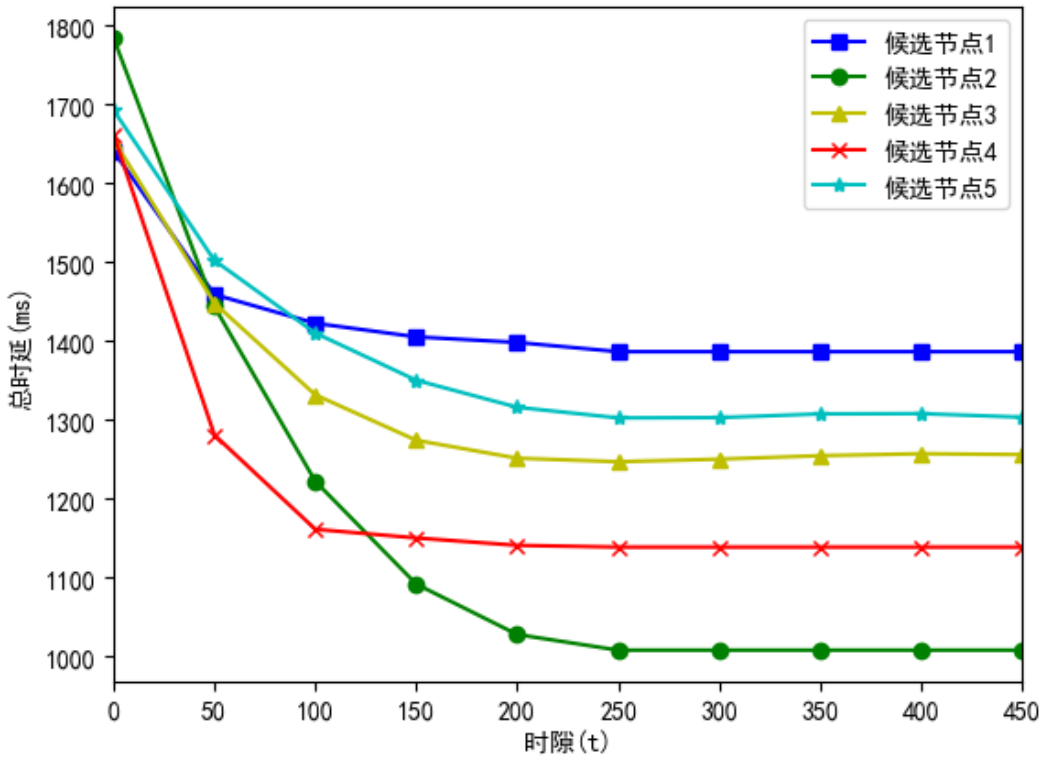

图 4.1　各候选节点的时延

由图 4.1 可知，对于候选集中的候选用户来说，每个用户对整个关系组的缓存服务能力有所差距。候选节点 1 和候选节点 5 的时延相差不多，但是和候选节点 2 的时延差距较大。这说明不同的节点有着不同的缓存服务能力。但是从候选节点 4 和候选节点 2 来看，虽然候选节点 4 的缓存空间比候选节点 2 的缓存空间大，但是候选节点

4 的总时延却比候选节点 2 的高。这证明了本文在一开始所提出的论点：用户的缓存服务能力与自身的缓存空间并不是绝对的依赖关系，选择缓存节点时有必要综合考量候选用户的自身设备条件、周围用户的位置和兴趣变化以及用户的请求等因素。对于不同候选节点来说，这在一定程度上说明了对于愿意成为缓存节点的候选用户有必要进行筛选来保证整个关系组的性能改善。从曲线走势来看，在刚开始训练的阶段，显示的整体时延较高，这是因为在刚开始训练阶段，候选节点中并没有缓存相关内容，导致普通用户在请求时命中率会比较低，只能从基站处得到内容。但是当训练多次后，候选用户周围的普通用户的兴趣偏好逐渐被候选用户掌握到并存储了相关内容，使得用户在请求同类型的内容的时候命中率逐渐上升，由于 D2D 通信速率较快，导致了整体时延在逐渐下降，最终在候选用户存储空间有限的情况下缓存了最有可能被请求的内容。上图说明了本文所提实时 MAB 算法的有效性。

由图 4.2 所示，我们在这里对各个候选节点之间的协作缓存进行了进一步验证。从该图中可以看到，我们对候选结果集中排名最前的节点当作主节点，比较了主节点和各个从节点之间的协作缓存对整个关系组的影响。从仿真结果来看，我们看到对不同的从节点由于他们的位置、自身存储设备以及周围用户的请求概率不一样，导致各个从节点在与主节点进行协作缓存时出现了对整个关系组的贡献有所差距。假设主节点与从节点随机协作缓存，虽然可以取得一定的收益，但是并不一定是使得整体时延达到最低的节点组合。比如多点组合模型 1 和多点组合模型 3 的协作缓存带来的收益并没有多点主从协作模型的收益高。从上图中可以了解到，我们提出的算法对降低用户请求时延具有良好的效果，但是各个组合之间有所差别，这也让我们看到了对整体贡献最多的协作缓存组合。

由图 4.3 可知，我们比较了 LRU 算法和 LFU 算法与本文所采用的算法。我们在主节点采用的是本文的基于实时 MAB 算法缓存更新模型，从仿真结果来看，实时 MAB 算法存在着一定的先进性。采用实时 MAB 算法的缓存节点为整个用户关系组带来的总时延比采用 LFU、LFU 等算法为整个用户关系组带来的总时延要低。说明我们基于此做研究是有着实际意义的。即本文提出的主从节点协作缓存模型是有依据可行的。除了对比实时 MAB 算法的有效性之外，我们还对比了多点随机选择策略，该策略主要是对于多个候选节点并不做有效区分直接拿来做缓存设备。从仿真结果看，多点随机选择策略的效果和主节点的效果相差无几。这说明了对于缓存节点的选择问题，并不能随意的选择，要有一定的实际根据，虽然取得了一定效果，但是离我们所要求的目标还有一定的距离，这也从侧面证明了本文所提算法的优化思路。多点独立缓存策略的核心是根据我们所做的对候选节点的筛选结果来建立多点分布式缓存，但是各个节点之间是单独工作的，缓存内容并不存在着联系，这种做法有自身的突出点是它的算法复杂度相对较低，而且还可以取得降低时延的作用。但是，这与我们追求

的尽可能降低整个关系组的请求时延的目标不相符，这是一方面。另一方面是对于多个独立分布的节点来说，它们的缓存内容存在着很多内容上的重复。在缓存空间有限的前提下，提高内容的多样性是需要进行改善的重点。由于各个节点之间的范围并不太远，它们之间可以进行协作，通过降低内容的重复性可以使得请求命中率得到提升，进而整个关系组整体的时延会得到进一步的降低。从仿真结果来看，多点独立缓存策略的效果比单个缓存节点为整个用户关系组带来的总时延要低，这也是进行多点缓存的根据。但是多点独立缓存的缺点也是很明显，从曲线上看多点独立缓存为整个用户关系组带来的时延比本文提出的主从节点协作缓存模型要高。这证实了我们分析时进行协作缓存的想法，同时也证明了我们提出的主从节点协作缓存算法取得了好的结果，证实了本文所提算法的有效性。

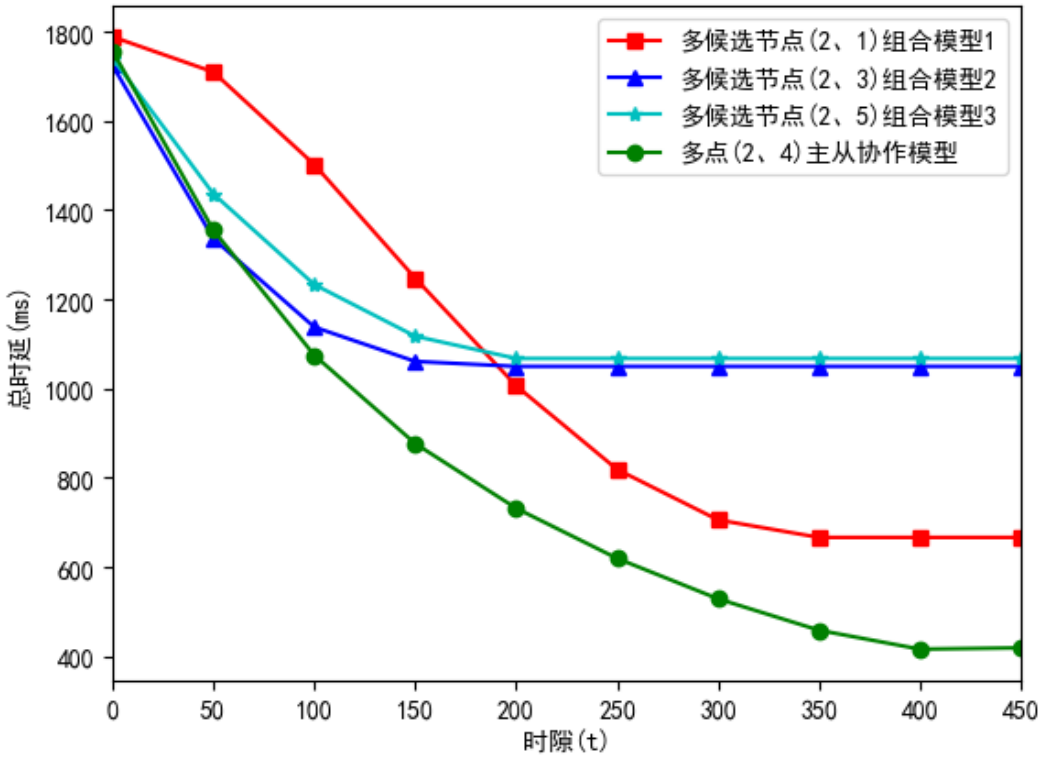

图 4.2　多候选节点之间的协作缓存

如图 4.4 所示，我们比较了对于所提出的实时 MAB 算法中所采用的探索和利用比例的不同对于整个关系组的时延影响。从图中曲线可以看出，不同的比率值对于整个模型收敛的速率影响有所差别。在探索占比较大时，我们可以在较少的次数下快速的找到某几类内容使得请求命中率提高，从而可以在缓存空间中缓存相应的内容。在探索占比较小时，由于我们在更可能多的利用已有知识进行动作选择，探索行为较少，使得我们发现能获得更多奖赏的内容的时间会靠后，收敛速度较慢，但这样相应的带

来了整体时延的下降更多的优势。本文选择三种探索和利用的比例来这说明选择一个合适的探索和利用的比例不仅对整个用户亲密关系组来说是一个重要的参数，而且对整个模型的实际应用具有重要的参考价值。在实际应用中，每个亲密用户关系组的情况不一致，比如用户的多少、组内用户的兴趣分布、用户的网络请求习惯等，都对探索和利用的比例产生重要的影响。

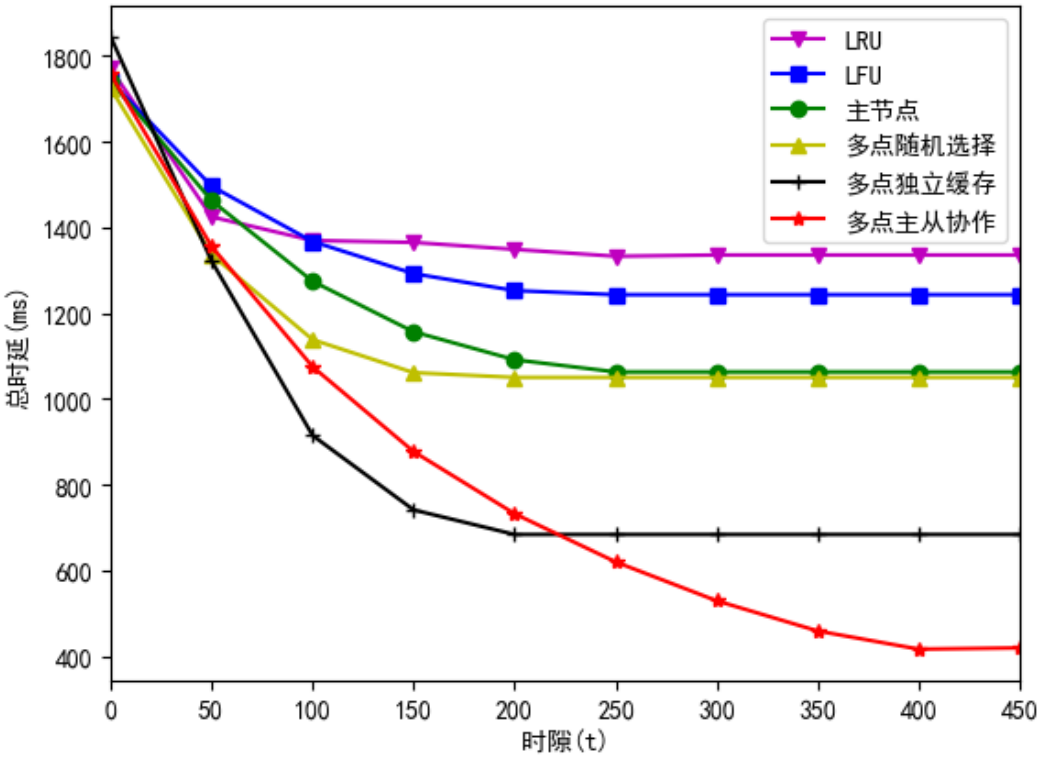

图 4.3 各方案的对比时延

如图 4.5 所示，我们在该图中比较了不同齐夫参数下对所提模型的影响。从图中可以看出，不同的参数下，齐夫参数为 0.3 和齐夫参数为 0.5 时对整个用户亲密关系组的影响差距并不太大，齐夫参数为 0.7 时对整个用户亲密关系组的影响比齐夫参数为 0.3 和齐夫参数为 0.5 时有较大不同，这也说明一个用户亲密关系组的用户的请求请求分布对模型的结果有着不同的影响，这样说明在设计不同亲密用户关系组时的缓存方案时根据各个组的实际情况分别进行调整对应的参数的重要性。同时从图形中也可以看出，不管当前用户对自身兴趣的请求是什么，随着训练的进行，所提出的模型总能有效的掌握周围用户的兴趣偏好，进而在相应的缓存中尽可能存储相应的内容。这从另一角度证实了本文所提的模型和算法的有效性。

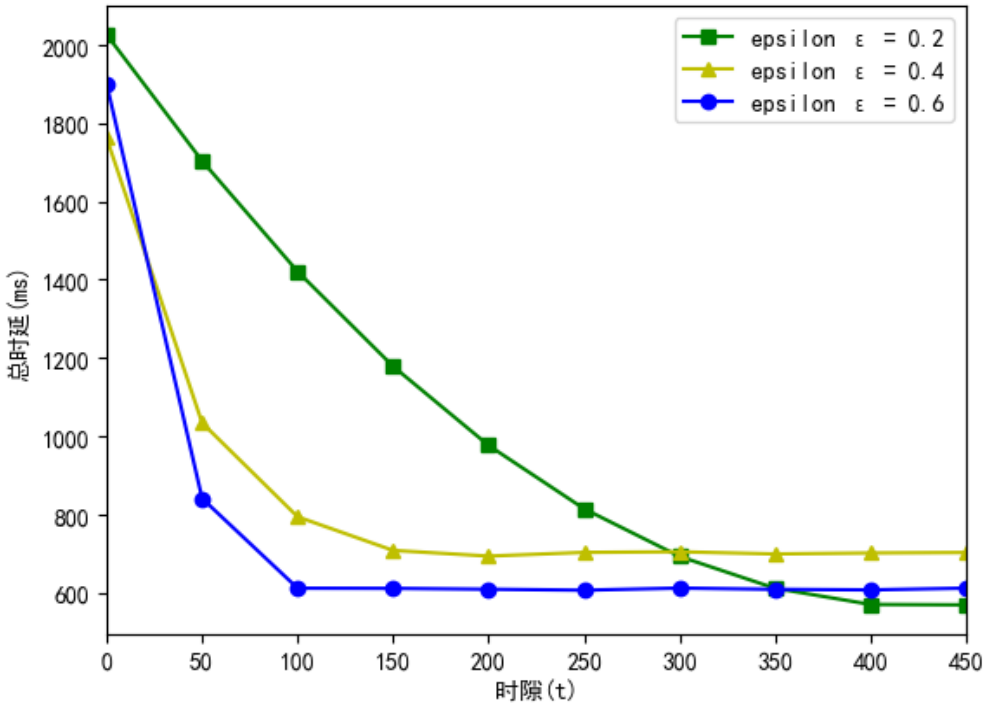

图 4.4　不同探索比例对时延的影响

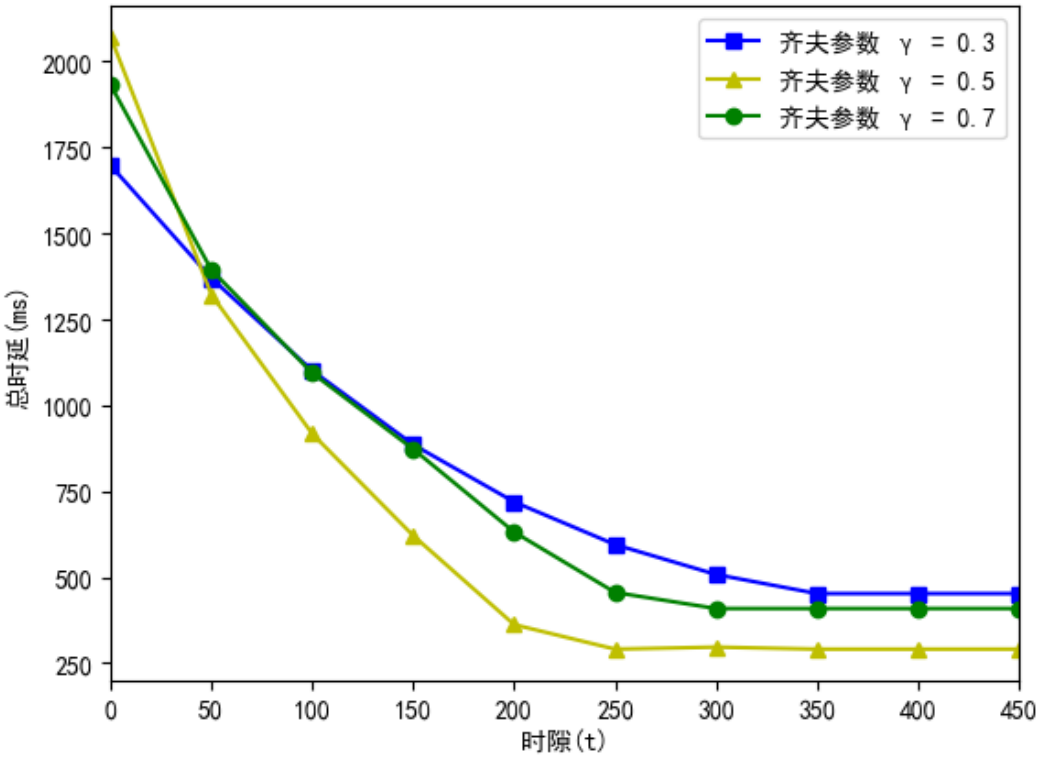

图 4.5　不同齐夫参数对时延的影响

4.4 移动条件下缓存节点的选择仿真结果

本小节主要为了测试本文所提的缓存节点选择算法的实际适用性。现实中的用户多具有移动性，测试采用静态条件下相同的用户组，主要区别是设定了用户可以进行移动，在移动中收集各个候选节点的缓存信息，并以此为依据进行缓存节点的选择。

本文主要采用了三组实验。分别对应的是只有普通用户进行移动、只有候选节点进行移动和候选节点与普通用户同时移动的情况。第一组实验用来测试候选节点周围的用户的移动对缓存节点选择的影响。第二组实验用来测试候选节点的移动对缓存节点选择的影响。第三组实验用来测试候选节点和周围的普通用户同时移动对缓存节点选择的影响。仿真结果指标主要包括用户组的总时延和累计奖赏。总时延用来标识候选用户在移动条件下是否可以选为缓存节点，累计奖赏用来从侧面验证被选为缓存节点的用户的确起到了降低请求时延的目的。综合三组实验我们可以验证本文所提算法是否具有实际适用性，为以后实际生活中应用此算法提供理论基础。

4.4.1 只有普通用户进行移动

本组仿真主要使发出请求的普通用户进行移动，以此来判断候选节点的缓存服务能力。仿真结果如图 4.6 和图 4.7。

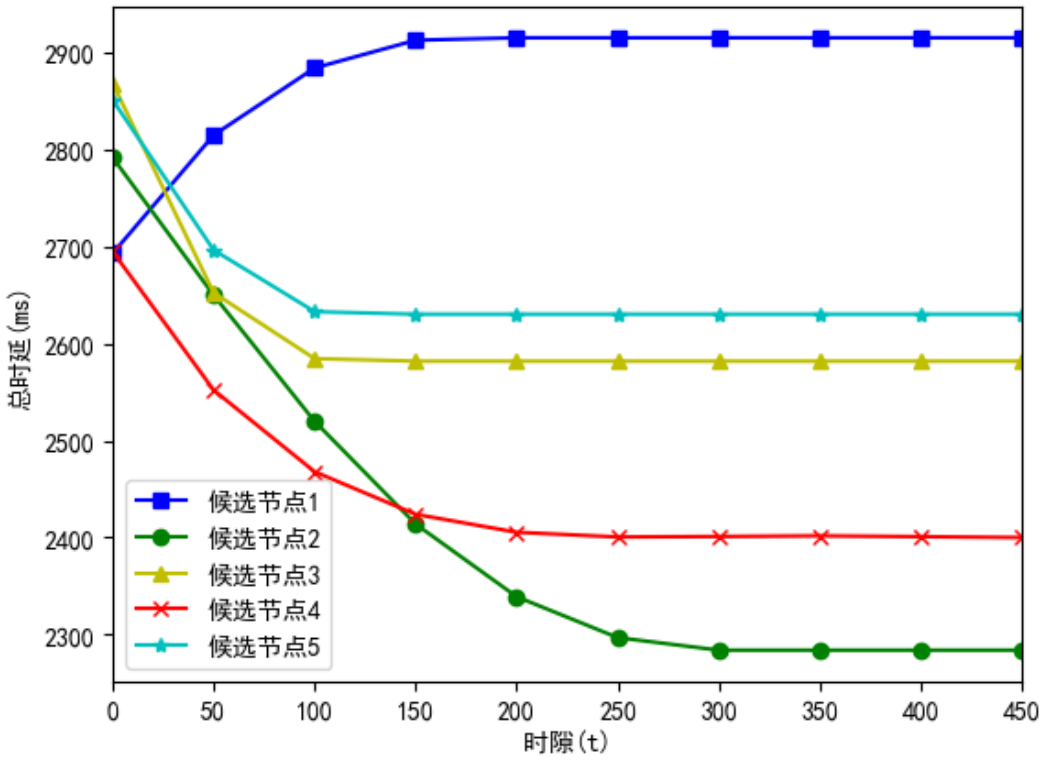

图 4.6 只有普通用户运动时各候选节点的时延

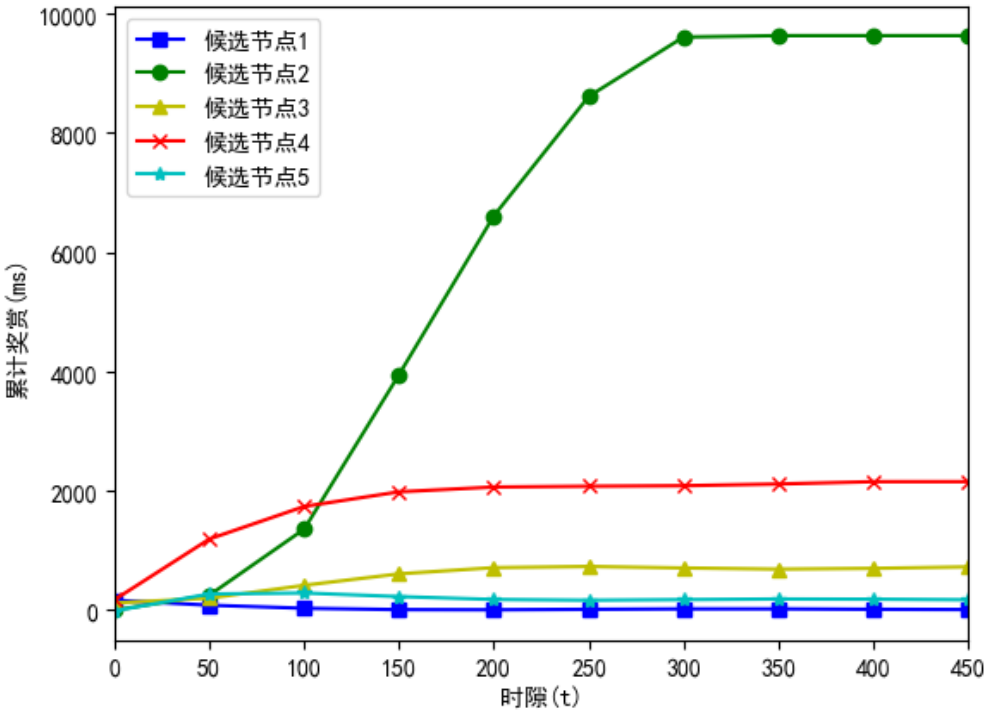

图 4.7　只有普通用户运动时各候选节点的累计奖赏

图4.6显示了在所有候选节点处于静止条件下周围普通用户的移动对用户请求的缓存服务能力。与静止条件下所有用户均不进行移动的差别主要在个别候选节点的缓存服务能力会随着周围用户位置的改变而相应的增强或降低，使得在选择缓存节点时做出的选择会有所不同。图 4.6 中候选节点 2 的总时延最低，在图 4.7 中候选用户 2 的累计奖赏最高。而本文设计的算法的奖赏值是时延降低量，时延降低量越大，奖赏值越大。这与仿真结果相符，验证了本文所提算法的实际适用性。而个别候选节点在周围用户的位置变化时，使得整个用户亲密关系组的总时延相应增加。比如候选节点 1 随着周围用户的移动，总时延在逐渐增高，其对应的累计奖赏也趋近于 0，说明了其已经不适合担任缓存节点。如果按照已有研究文献中提到的仅仅根据用户的缓存设备能力和历史请求信息来确定缓存节点，在用户移动条件下，虽然该用户可以与个别用户可以建立通信连接来分享内容，但是却对整个用户关系组的性能没有实际的提升。这也证实了缓存节点的选择不应该单单只考虑候选节点本身的能力，还应该考虑其与周围用户的位置和请求信息等因素的联系。

4.4.2　只有候选节点进行移动

本组仿真使发出请求的普通用户保持静止，而让各候选节点进行移动，来考量候选节点在移动条件下的缓存服务能力。仿真结果如图 4.8 和 4.9。

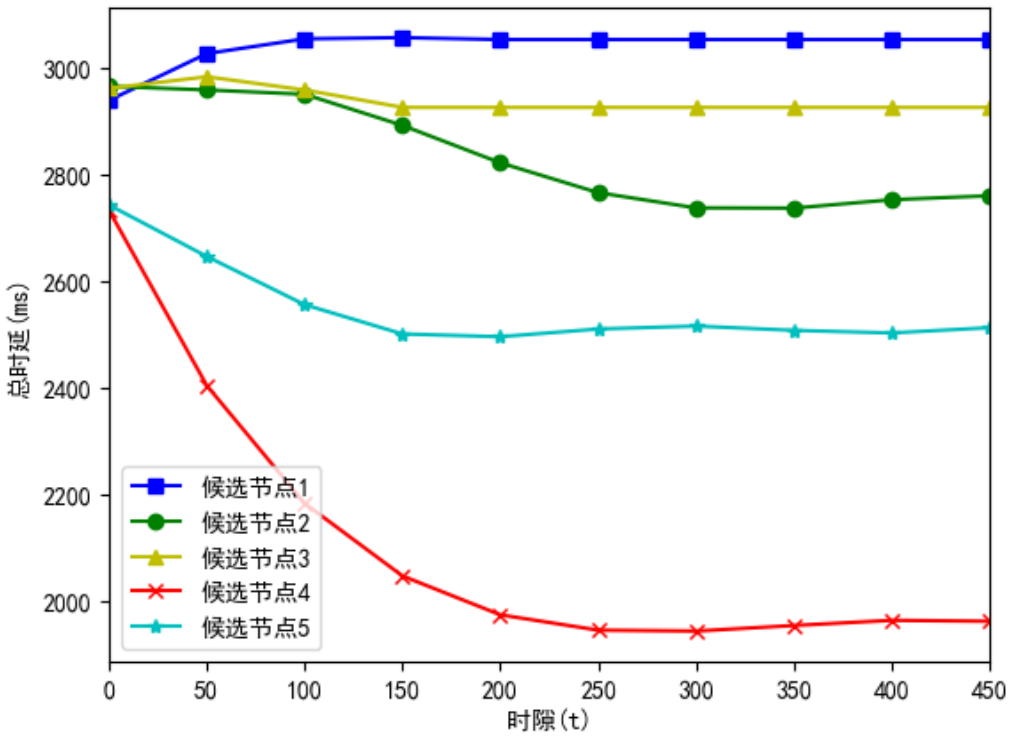

图 4.8 只有候选节点运动时各候选节点的时延

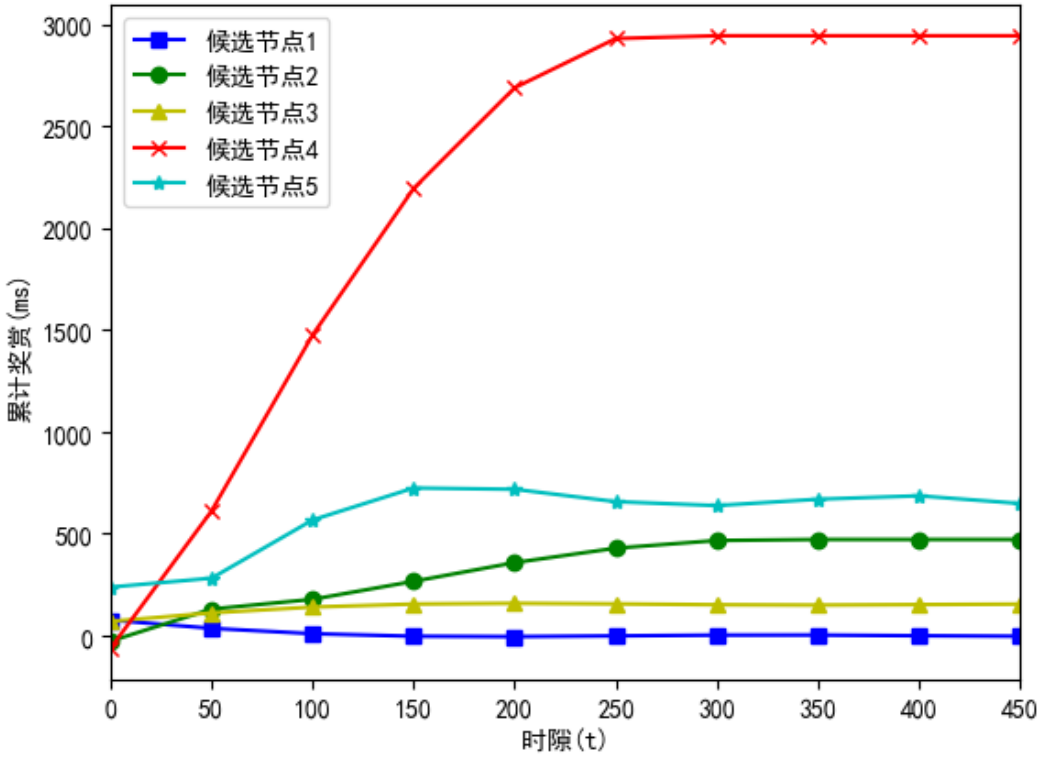

图 4.9 只有候选节点运动时各候选节点的累计奖赏

图 4.8 显示了本文所提算法在只有候选节点进行移动时各候选节点的缓存服务能力。可以看出候选节点 2 不再是使得整体时延降低最小的候选节点。候选节点 4 的总时延在训练后达到最低。并且从图 4.9 中可以看出的候选节点 4 的累计奖赏也是最高，这说明了本文所提算法具有实际适用性。但是，候选节点的移动导致多数其他候选节点为整个亲密关系组带来的总时延并没有明显的降低，反倒是出现了不同程度的上升。这说明了候选节点的移动对整个用户亲密关系组的影响较大。在实际应用中设计实际模型时应该着重考虑这一因素，可以尝试选择那些运动的范围并不是很大的用户作为缓存节点。这也从侧面说明了缓存节点的选择并不是一成不变的。其需要记录各个节点的位置和请求信息来综合评定。这对实际中进行缓存节点的协作缓存具有重要的参考价值。

4.4.3 候选节点与普通用户同时移动

本组仿真考虑了个别候选节点和普通用户同时进行移动时的因素。仿真结果如图 4.10 和图 4.11。

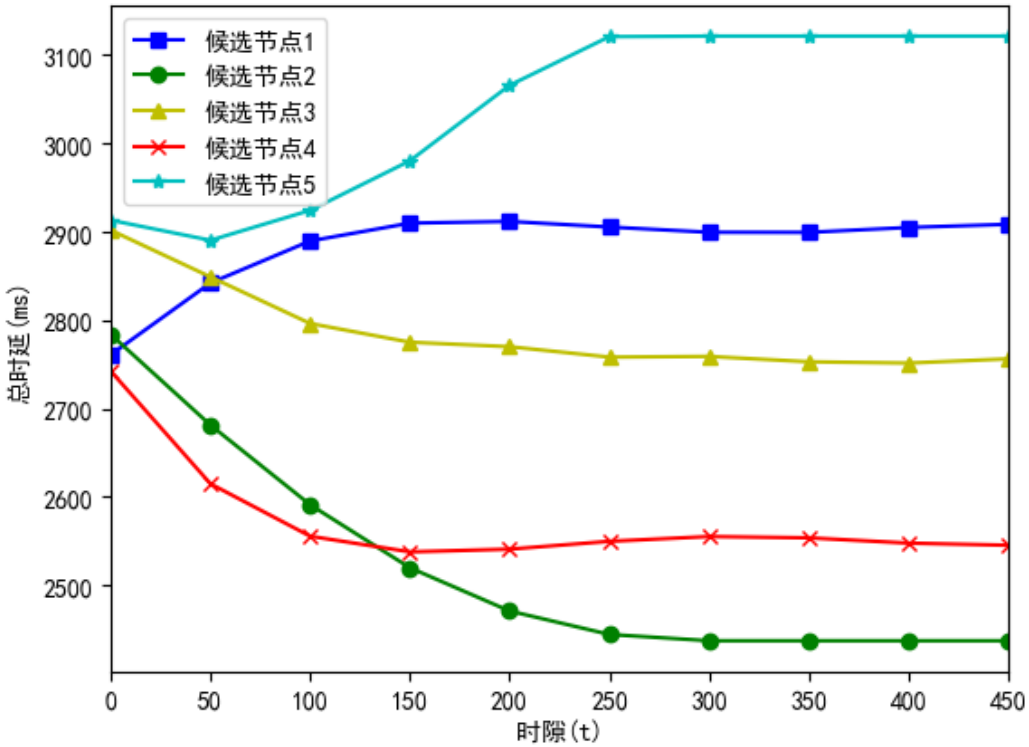

图 4. 10 候选节点和普通用户同时运动时各候选节点的时延

图 4.10 和图 4.11 的结果和只有普通用户进行移动时的结果大致相同。不同点在于进行移动的候选节点 5 的总时延随着训练的增加不断上升，这显示了在候选节点进

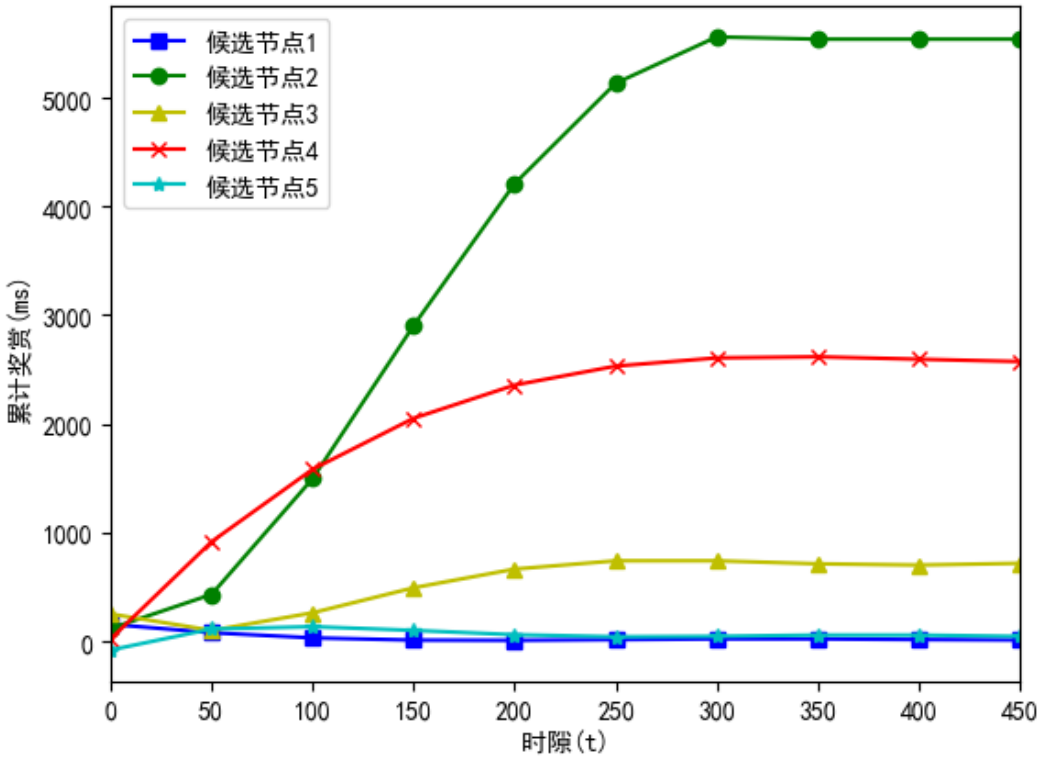

图 4.11 候选节点和普通用户同时运动时各候选节点的累计奖赏

行移动时对整体的服务能力会有较大影响，也影响了缓存节点的选择。但是这也说明了当候选节点和普通用户同时进行移动时对缓存节点的选择具有一定的随机性。当用户以不同于往日的规律进行移动时，会对整个用户关系组的缓存节点选择带来影响。用户的移动不确定性使得缓存节点在做缓存决策时需要实时做出改变，这也是以强化学习为基础进行算法设计的目的。

4.5 本章小结

本章主要对第三章提出的缓存节点算法和主从节点协作缓存模型进行仿真验证。首先在静止条件下验证了各个候选节点的缓存服务能力，并以此为依据进行缓存节点选择。接着验证了各个候选节点协作缓存的服务能力，可以看出进行缓存节点选择的必要性。并与 LFU、LRU 等方案进行了对比。最后在在移动性条件下测试了提出的缓存节点选择算法，主要采取了三组仿真来验证该算法。最后仿真结果证实了所提模型的有效性和实际适用性。

第五章 总结与展望

5.1 总结

随着信息技术的发展，人们对于网络的热情愈发强烈，网络中的数据流量呈现爆发式增长，尤其是时下视频等网络资源的快速增长，对当下网络结构提出了更高的要求。爆发式的流量使得网络变得拥挤，降低了人们的上网体验，D2D 缓存技术作为5G 时代的新兴技术是一个研究热点。本文在文章一开始交代了 D2D 缓存技术的背景与意义，详细描述了 D2D 缓存技术对于现阶段的网络优化的重要性。接着我们交代了国内和国外在此方向上的研究进展，同时也说明了已有研究的不足之处。现阶段的研究主要集中在如何设计一个有效的缓存机制来优化网络结构，减轻网络拥塞。大多数研究者的方向主要为利用用户之间的社会关系、评估缓存内容流行度、协作缓存等。本文根据文献中已经提到的方法，提出了用户亲密关系组的概念。这一点主要来源于已有研究对用户的社会关系的研究。用户之间的社会关系使得用户之间拥有了一定的信任值，在此基础上可以进行 D2D 通信。现实生活中存在的众多亲密关系组符合用户之间具有信任值的条件。因此，在亲密关系组的基础上作进一步的研究是有现实意义的。

本文针对已有缓存机制中的不足之处，提出了自己的解决方案。本文从降低网络回程拥挤度和提高延迟降低量的角度，结合当下热门的强化学习技术，提出了缓存节点选择算法。该算法不同于已有的节点选择算法，该算法主要考虑了缓存节点自身存储能力和周围的用户请求分布的不同，根据周围用户的实时请求来决定节点的缓存服务能力，进一步得到在整个关系组内缓存服务能力较强的缓存节点。主要的方式是采用基于多臂赌博机模型来优化缓存决策，降低缓存节点周围用户请求的整体时延，选择使得整个用户关系组的整体时延较低的缓存节点。这里得到了缓存节点的缓存服务能力列表，在这里可以根据当前用户关系组的人数组成来决定选择移动数量的缓存节点。此时各个缓存节点是相互独立的。在已有研究中，研究者们通常会选择节点之间的协作来达到更优的结果。但是不同缓存节点的缓存服务能力有所差异，本文为了充分利用各缓存节点的服务能力，本文提出了一种主从节点协作缓存模型。此模型主要思路是充分利用缓存服务能力强的节点，使得缓存服务能力强的节点之间可以进一步的协作，做出有益于整体用户亲密关系组的缓存决策。该模型进一步降低了缓存节点服务区域内的整体时延，使得各缓存节点得到了充分利用。而且该模型利用了节点选择时的缓存决策，提高了算法的收敛速度，并准确的把握了周围用户的兴趣爱好。最后为了验证本文所提算法的实际适用性，本文从实际应用场景出发，采用了随机游动

模型来实际验证本文提出的节点选择算法的实际适用性。从仿真结果来看，本文所提算法展现出了较好的性能。

5.2 展望

本文接下来的研究方向主要有两个。第一，继续完善本文所提的算法和模型。本文在某些细节上处理的不够完美，需要进一步的处理。同时希望可以推动实际系统的搭建测试。本文所提算法虽然在仿真上符合了理论预期，但是需要实际推动建立采用所提算法和模型的系统。如在实际系统中可以取得不错的效果，则将会增强继续科研的信心和动力，毕竟理论与实践相结合才能使得研究工作得到更多的肯定。第二，从用户亲密关系组的角度，深入探索不同亲密关系组和整个大区域范围内的协作缓存。此举主要考虑了部分与整体的协作关系，各部分的用户情况的不一致性是需要重点关注的变量，在整体协作缓存时也需要关注那些自愿与陌生人建立通信连接的因素，这会为大区域协作缓存带来补充，使得区域缓存决策可以更加符合区域用户的需求，进一步提高网络结构的稳定性和用户满意度。

参考文献

[1] 王喜瑜，向际鹰.专题:5G 商用支撑理论及关键技术[J].中兴通讯技术，2019，25(01):5.

[2] 祝晓悦．蜂窝网络下 D2D 通信性能的研究[D]．北京邮电大学，2013.

[3] 王帆．5G 关键技术 D2D 的相关研究[D]．北京交通大学，2016.

[4] Na D, Martin H. The Benefits of Hybrid Caching in Gauss-Poisson D2D Networks[J]. IEEE Journal on Selected Areas in Communications, 2018:1-1.

[5] Li J, Liu M, Lu J, et al. On Social-Aware Content Caching for D2D-Enabled Cellular Networks with Matching Theory[J]. IEEE Internet of Things Journal, 2017:1-1.

[6] Xu C, Gao C, Zhou Z, et al. Social Network-Based Content Delivery in Device-to-Device Underlay Cellular Networks Using Matching Theory[J]. IEEE Access, 2016, PP(99):1-1.

[7] Ma C, Ding M, Chen H, et al. Socially Aware Caching Strategy in Device-to-Device Communication Networks[J]. IEEE Transactions on Vehicular Technology, 2018:1-1.

[8] Li J, Liu M, Lu J, et al. On Social-Aware Content Caching for D2D-Enabled Cellular Networks with Matching Theory[J]. IEEE Internet of Things Journal, 2017:1-1.

[9] 黄桔林，卢旭文．5G 关键技术：D2D 通信技术应用[J]．电子技术与软件工程，2019.

[10] 王莹，费子轩，张向阳，et al．移动边缘网络缓存技术[J]．北京邮电大学学报，2017.

[11] Zhu K, Zhi W, Zhang L, et al. Social-Aware Incentivized Caching for D2D Communications[J]. IEEE Access, 2016, 4:7585-7593.

[12] Wu D, Yan J, Wang H, et al. Social Attribute Aware Incentive Mechanism for Device-to-Device Video Distribution[J]. IEEE Transactions on Multimedia, 2017:1-1.

[13] 贾庆民．5G 移动通信网络中缓存与计算关键技术研究[D]．北京邮电大学，2019.

[14] W. Jiang, G. Feng, S. Qin, et al. Multi-Agent Reinforcement Learning for Efficient Content Caching in Mobile D2D Networks[J]. IEEE transactions on wireless communications, 18(3):1610-1622.

[15] Somuyiwa S O, Gyorgy A, Gunduz D. A Reinforcement-Learning Approach to Proactive Caching in Wireless Networks[J]. IEEE Journal on Selected Areas in Communications, 2017.

[16] Muller S, Atan O, Mihaela V D S, et al. Context-Aware Proactive Content Caching With Service Differentiation in Wireless Networks[J]. IEEE Transactions on Wireless Communications, 2017, 16(2):1024-1036.

[17] Bharath B N, Nagananda K G, Poor H V. A Learning-Based Approach to Caching in Heterogenous Small Cell Networks[J]. IEEE Transactions on Communications, 2016, 64(4):1674-1686.

[18] Li W, Wang J, Zhang G, et al. A Reinforcement Learning Based Smart Cache Strategy for Cache-Aided Ultra-Dense Network[J]. IEEE Access, 2019:1-1.

[19] Sutton R, Barto A. Reinforcement Learning: An Introduction[M]. MIT Press, 1998.

[20] Sengupta A, Amuru S D, Tandon R, et al. Learning distributed caching strategies in small cell networks[C]. 2014 11th International Symposium on Wireless Communications Systems (ISWCS). IEEE, 2014.

[21] Bastug E, Bennis M, Debbah, Mérouane. Living on the edge: The role of proactive caching in 5G wireless networks[J]. IEEE Communications Magazine, 2014, 52(8):82-89.

[22] 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.

[23] Gregori M, Jesús Gómez-Vilardebó, Matamoros J, et al. Wireless Content Caching for Small Cell and D2D Networks[J]. IEEE Journal on Selected Areas in Communications, 2016, 34(5):1222-1234.

[24] M. Afshang, H. S. Dhillon, P. H. J. Chong. Fundamentals of Cluster-Centric Content Placement in Cache-Enabled Device-to-Device Networks[J]. IEEE Transactions on Communications, 2016, 64(6): 2511-2526.

[25] Liu Xiaonan, Zhao Nan, Chen, Yunfei, et al. Dense D2D-Connection Establishment via Caching in Small-Cell Networks[C]. 2018 24th Asia-Pacific Conference on Communications (APCC). IEEE, 2018.

[26] Bai B, Wang L, Han Z, et al. Caching based socially-aware D2D communications in wireless content delivery networks: a hypergraph framework[J]. IEEE Wireless Communications, 2016, 23(4):74-81.

[27] Ahmed E, Yaqoob I, Gani A, et al. Social-Aware Resource Allocation and Optimization for D2D Communication[J]. IEEE Wireless Communications, 2017:2-9.

[28] Jameel F, Hamid Z, Jabeen F, et al. A Survey of Device-to-Device Communications: Research Issues and Challenges[J]. Communications Surveys & Tutorials, IEEE, 2018, 20(3):2133-2168.

[29] Tang H, Ding Z. Mixed Mode Transmission and Resource Allocation for D2D Communication[J]. Wireless Communications IEEE Transactions on, 2016, 15(1):162-175.

[30] Chou H J, Chang R Y. Joint Mode Selection and Interference Management in Device-to-Device Communications Underlaid MIMO Cellular Networks[J]. IEEE Transactions on Wireless Communications, 2017, 16(2):1120-1134.

[31] 徐征. 关于面向 5G 通信网的 D2D 技术阐述[J]. 2018.

[32] 牛煜霞. D2D 辅助蜂窝网络中的协作缓存策略[D]. 北京邮电大学, 2019.

[33] 周志华, 机器学习, 清华大学出版社, 2016.

[34] P. Dorato. Dynamic programming and stochastic control[J]. IEEE Transactions on Systems Man & Cybernetics, 1978, 23(5):967-968.

[35] Ben J.A. Kröse. Learning from delayed rewards[J]. Robotics & Autonomous Systems, 1995,

15(4):233-235.

[36] Sutton R S. Learning to Predict by the Methods of Temporal Differences[J]. Machine Learning, 1988, 3(1):9-44.

[37] Wang F Y, Zhang H, Liu D. Adaptive Dynamic Programming: An Introduction[J]. IEEE Computational Intelligence Magazine, 2009, 4(2):39-47.

[38] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou. Playing Atari with Deep Reinforcement Learning. NIPS 2013.

[39] Muller S, Atan O, Mihaela V D S, et al. Context-Aware Proactive Content Caching With Service Differentiation in Wireless Networks[J]. IEEE Transactions on Wireless Communications, 2017, 16(2):1024-1036.

[40] Bharath B N, Nagananda K G, Poor H V. A Learning-Based Approach to Caching in Heterogenous Small Cell Networks[J]. IEEE Transactions on Communications, 2016, 64(4):1674-1686.

[41] X. Xu, M. Tao, Collaborative Multi-Agent Reinforcement Learning of Caching Optimization in Small-Cell Networks[C]. 2018 IEEE Global Communications Conference (GLOBECOM). Abu Dhabi, United Arab Emirates: IEEE, 2018.

[42] Gregori M, Jesús Gómez-Vilardebó, Matamoros J, et al. Wireless Content Caching for Small Cell and D2D Networks[J]. IEEE Journal on Selected Areas in Communications, 2016, 34(5):1222-1234.

[43] S. Kim, E. Go, Y. Song, et al. A Study on D2D Caching Systems with Mobile Helpers[C]. 2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN), Prague: IEEE, 2018.

致谢

首先要感谢生我养我的父母，没有他们就没有我的今天。

三年的研究生生涯即将结束，在这里要感谢陈健老师在这三年里的耐心指导。不管是科研还是生活，陈老师亦师亦友，经历过不少次的迷茫，在陈老师的聆听与教诲下慢慢走出了困境，也明白了今后的路应该如何去闯荡。其次，要感谢吕璐师兄与张威师兄在期刊论文写作时候的细心的修改与建议，经历了这个过程才明白科研过程的严谨性，为今后做类似工作积攒了经验。接着要感谢任琦琦师姐在科研道路上的指点，在每次的周会与论文的阅读等方面给予了很多的帮助。然后要感谢舍友侯雨君、黄露和闫航以及实验室的同学，感谢他们三年来的陪伴，无数次的互黑与胡吹，无数次的聚餐使得在枯燥的学习与科研生涯中有了更多前进的动力，我们一起见证入学，又将一起相伴毕业，谢谢你们的青春，祝愿你们今后身体健康、工作顺利、事业有成、抱得美人归。

最后，要感谢一直陪我的女票漫漫同学，在我心情低沉的时候，你总是乐观的开导我，跟你在一起感觉生活轻松了许多，世上根本没有什么难事。谢谢你一直在，希望我们一直走下去。

作者简介

1. 基本情况

轩夺，男，河南商丘人，1996 年 5 月出生，西安电子科技大学通信工程学院通信与信息系统专业 2017 级硕士研究生。

2. 教育背景

2013.08～2017.07 西安电子科技大学，本科，专业：通信工程
2017.08～ 西安电子科技大学，硕士研究生，专业：通信与信息系统

3. 攻读硕士学位期间的研究成果

3.1 发表学术论文

[1] Hang Yan, Zhan Yan, Duo Xuan, et al. Denoising Framework Based on External Prior Guided Rotational Clustering[J]. IET Image Processing. (已录用未发表)

3.2 参与科研项目及获奖

[1] 跳频信号的盲识别, 2017.11-2019.06, 已完成, 负责频率识别和码元速率估计。