



研究与开发

机器类通信中集中式与分布式 Q 学习的资源分配算法研究

余云河, 孙君

(南京邮电大学通信与信息工程学院, 江苏 南京 210023)

摘要: 针对海量机器类通信 (massive machine type communication, mMTC) 场景, 以最大化系统吞吐量为目标, 且在保证部分机器类通信设备 (machine type communication device, MTCD) 的服务质量 (quality of service, QoS) 要求前提下, 提出两种基于 Q 学习的资源分配算法: 集中式 Q 学习算法 (team-Q) 和分布式 Q 学习算法 (dis-Q)。首先基于余弦相似度 (cosine similarity, CS) 聚类算法, 考虑到 MTCD 地理位置和多级别 QoS 要求, 构造代表 MTCD 和数据聚合器 (data aggregator, DA) 的多维向量, 根据向量间 CS 值完成分组。然后分别利用 team-Q 学习算法和 dis-Q 学习算法为 MTCD 分配资源块 (resource block, RB) 和功率。吞吐量性能上, team-Q 和 dis-Q 算法相较于动态资源分配算法、贪婪算法分别平均提高了 16%、23%; 复杂度性能上, dis-Q 算法仅为 team-Q 算法的 25% 及以下, 收敛速度则提高了近 40%。

关键词: 资源分配; 集中式 Q 学习; 分布式 Q 学习; 余弦相似度; 多维向量

中图分类号: TP929.5

文献标识码: A

doi: 10.11959/j.issn.1000-0801.2021244

Research on resource allocation algorithm of centralized and distributed Q-learning in machine communication

YU Yunhe, SUN Jun

College of Telecommunications and Information Engineering, Nanjing University of Posts and
Telecommunications, Nanjing 210023, China

Abstract: Under the premise of ensuring partial machine type communication device (MTCD)'s quality of service (QoS) requirements, the resource allocation problem was studied with the goal of maximizing system throughput in the massive machine type communication (mMTC) scenario. Two resource allocation algorithms based on Q-learning were proposed: centralized Q-learning algorithm (team-Q) and distributed Q-learning algorithm (dis-Q). Firstly, taking into account MTCD's geographic location and multi-level QoS requirements, a clustering algorithm based on cosine similarity (CS) was designed. In the clustering algorithm, multi-dimensional vectors that represent MTCD and data aggregator (DA) were constructed, and MTCDs can be grouped according to the CS value between multi-dimensional vectors. Then in the MTC network, the team-Q learning algorithm and dis-Q learning algorithm were

收稿日期: 2021-04-30; 修回日期: 2021-10-20

通信作者: 孙君, sunjun@njupt.edu.cn

基金项目: 国家自然科学基金资助项目 (No.61771255); 中国科学院重点实验室开放课题 (No.20190904)

Foundation Items: The National Natural Science Foundation of China (No.61771255), Open Project of Key Laboratory of Chinese Academy of Sciences (No.20190904)



used to allocate resource blocks and power for the MTCD. In terms of throughput performance, team-Q and dis-Q algorithms have an average increase of 16% and 23% compared to the dynamic resource allocation algorithm and the greedy algorithm, respectively. In terms of complexity performance, the dis-Q algorithm is only 25% of team-Q algorithm and even below, the convergence speed is increased by nearly 40%.

Key words: resource allocation, centralized Q-learning, distributed Q-learning, cosine similarity, multi-dimensional vector

1 引言

机器型通信 (machine type communication, MTC) 允许智能物体在没有人干预情况下实现相互通信, 3GPP (3rd Generation Partnership Project) 认为 MTC 将会对物联网 (internet of things, IoT) 的发展起到关键作用^[1-2]。随着 IoT 的普及, 对“物”之间的通信具有很高的需求, 即使 5G 也不能保证满足未来新业务的所有需求, 因此在 B5G (beyond 5G) 和 6G 网络中, MTC 将会是研究人员关注的重点^[3-4]。思科预测到 2022 年各行业中将会有 39 亿个 MTC 设备连接到网络中^[5], 而海量机器类通信设备 (machine type communication device, MTCD) 连接不仅导致频谱资源匮乏, 还会造成网络拥塞, 给基站 (base station, BS) 带来沉重负担。

在部署了高密度 MTCD 的 mMTC 网络中, 将 MTCD 分组为较小集群被视为一种有助于缓解 MTC 网络拥塞, 提高 MTCD 接入成功率, 进而促进吞吐量提升的技术^[6]。为此, 国内外学者提出了一系列关于 MTCD 分组聚类算法。文献[6-7]分别依据设备的 QoS 要求、地理位置进行分组, 文献[8]为了延长网络寿命, 依据 MTCD 剩余能量以及与 BS 间的距离进行聚类。文献[9-10]在传统 *k*-means 算法基础上作出改进, 分别针对 MTCD 能量效率与 MTC 网络传输时延要求, 对 MTCD 进行聚类。然而, 上述研究中提出的 MTCD 聚类策略, 有的仅考虑了地理位置和 QoS 要求中的单个因素, 并未充分发掘 MTCD 之间的关联性, 导致在 MTCD 聚簇内不能很好地协调干扰, 潜在

影响系统吞吐量; 有的仅针对特定优化目标进行聚类, 不具有普遍适用性。

文献[11-12]均考虑 H2H (human to human) 与 M2M (machine to machine) 共存场景中系统用户过载情况下的资源分配问题。然而, 文献[11]未考虑时延敏感 M2M 业务的传输需求, 导致无法满足此类 M2M 业务的 QoS 要求, 文献[12]则利用基于背包模型的资源分配算法, 保证了时延敏感 M2M 通信业务的 QoS, 但在文献[12]中仅将所提算法同传统的优先为 H2H 终端分配资源的算法进行性能比较, 无法充分验证该算法的优越性。文献[13]提出了一种动态资源分配策略用于解决 MTCD 间的资源分配问题, 虽然考虑了 MTCD 请求过载的情况, 但并不允许资源复用, 导致频谱利用率较低, 同时由于接入网络的 MTCD 数量较少, 也造成系统吞吐量下降。文献[14]针对多输入多输出系统中动态资源分配问题, 提出了一种确保用户最低 QoS 要求的资源分配算法, 能获得较高的系统吞吐量, 然而该方法是在用户功率等分配的前提下执行的, 并不符合实际, 具有一定的局限性。文献[15]讨论了在频谱资源匮乏条件下, 基于设备到设备 (device to device, D2D) 分簇的车通信资源分配问题, 在保证车用户正常通信下, 最大化蜂窝用户的吞吐量。文献[16]研究了基于容量最大化地 mMTC 场景的资源分配问题, 但使用的是传统粒子群算法, 该算法对容量提升作用有限, 且没有考虑 MTCD 分组问题。在功率有限、频谱资源匮乏的 MTC 网络中, 传统资源分配方法难以满足 MTCD 不断增长的 QoS 要求。近年来研究表明基于机器学习的资源分配策略已经优于传

统的方法^[17-18], 而 Q 学习作为一种著名无模型强化学习 (reinforcement learning, RL) 算法引起了人们的关注。

基于以上分析, 本文在确保承担高信噪比传输任务的 MTCD 最低 QoS 要求前提下, 提出两种 Q 学习算法: team-Q 学习算法和 dis-Q 学习算法, 解决网络内 MTCD 之间的资源块和功率联合分配问题。该资源分配算法分为两个阶段: 第一阶段设计一种基于 CS 的聚类方案, 即借鉴商品推荐系统中求取用户之间相似度的做法, 分别为 MTCD、DA 构造多维向量, 再利用向量之间余弦相似度进行分组。第二阶段中, 针对分组后的 MTC 网络上链路资源块和功率分配问题, 提出了两种基于 Q 学习的分配算法: team-Q 学习和 dis-Q 学习, 其中 dis-Q 算法在 team-Q 算法基础上改进了 Q 值表和奖励函数。最后, 通过仿真验证了所提算法能在复杂性、收敛速度以及对系统吞吐量促进作用等方面的有效性。

2 系统模型

本文研究的系统模型如图 1 所示, 随机分布的 MTCD 经过聚类后形成 MTCD 聚簇, 每个聚簇内含有一个数据聚合器 DA, 构成 MTC 网络。在 MTC 网络中, MTCD 通过稀疏码分多址技术^[19]与 DA 连接, DA 充当数据接收和转发的角色, 即负责接收 MTCD 数据并转发至 BS, 使得整个网络变成双层架构, 可以减轻 BS 的接入负担。假定聚簇与聚簇间使用正交的频谱资源, 而聚簇内的 MTCD 之间以非正交多址方式共用资源块。因此, 在 MTC 网络内由于资源块的复用会产生多址干扰, 在接收端则可采用串行干扰消除 (successive interference cancellation, SIC) 技术进行正确解调。

现实系统中通常会存在多种业务类型, 且不同业务类型间的 QoS 要求差异较大。为了体现服务差异化, 在 MTC 网络内部署了承担不同传输任务的 MTCD, 为此在 MTC 网络中设定 h

个最低信干噪比 (signal-to-interference-plus-noise ratio, SINR) 等级, 并使用 SINR_{th} 表示。具体地, SINR_{th} 表示为: $\text{SINR}_{\text{th}} = \{\text{SINR}_{\text{th},0}, \text{SINR}_{\text{th},1}, \dots, \text{SINR}_{\text{th},t}, \dots, \text{SINR}_{\text{th},h-1}\}$, SINR_{th} 为升序排列, 且 $\text{SINR}_{\text{th},0} > 0$, 表示网络中对信噪比要求最低的传输业务。

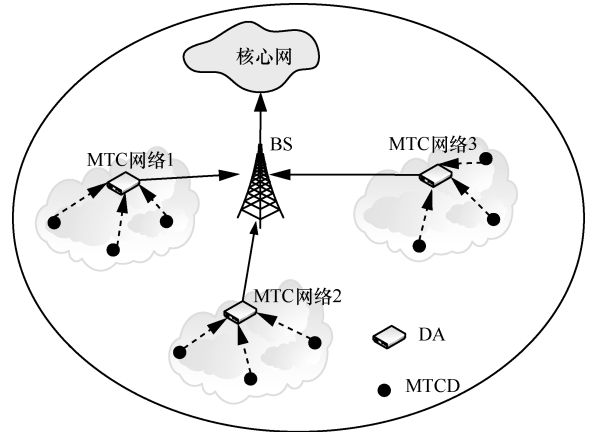


图 1 系统模型

设定在 MTC 网络中共有 K 个 RB、 M 个 MTCD、 N 个 DA, 使用 $S_{\text{RB}} = \{\text{RB}_1, \text{RB}_2, \dots, \text{RB}_k, \dots, \text{RB}_K\}$ 、 $S_{\text{MTCD}} = \{\text{MTCD}_1, \text{MTCD}_2, \dots, \text{MTCD}_m, \dots, \text{MTCD}_M\}$ 、 $S_{\text{DA}} = \{\text{DA}_1, \text{DA}_2, \dots, \text{DA}_n, \dots, \text{DA}_N\}$ 表示, $M \geq K$ 。其中, 集合 S_{MTCD} 由两个子集合 S_1 、 S_2 组成, S_1 中 MTCD 的最低信干噪比要求是 $\text{SINR}_{\text{th},0}$, 意味着不具有任何 QoS 要求, S_2 中的 MTCD 具有除 $\text{SINR}_{\text{th},0}$ 之外的其他 $h-1$ 个级别的 QoS 要求。为了提高频谱利用率, 预先为 S_2 中的 MTCD 分配资源块, 每个 MTCD 限定分得 1 个资源块, 并在保证 S_2 中 MTCD 的最低信干噪比要求前提下, 允许 S_1 中的 MTCD 共用 S_2 中的资源块, 每个 MTCD 可以分得多个资源块。为了描述方便, 在下文中分别使用 MTCD_m 、 MTCD_i 、 MTCD_j 指代 S_{MTCD} 、 S_1 、 S_2 中的 MTCD, $m=1, 2, \dots, M$ 、 $i=1, 2, \dots, |S_1|$ 、 $j=1, 2, \dots, |S_2|$, $|S_1|$ 、 $|S_2|$ 分别为 S_1 、 S_2 中 MTCD 数量, 且 $|S_1| + |S_2| = M$ 。

假定系统中所有链路经历慢衰落和路径损耗。因此, S_1 中的 MTCD_i 到数据聚合器 DA_n 之



间的瞬时链路增益为:

$$g_{i,n} = G \times \Gamma_{i,n} \times d_{i,n}^{-\alpha} \quad (1)$$

其中, G 、 $\Gamma_{i,n}$ 、 α 、 $d_{i,n}$ 分别为路径损耗常数、服从对数正态分布的慢衰落、路径损耗指数以及 MTCD_{*i*} 与 DA_{*n*} 之间的距离。同理也可定义 S_2 中的 MTCD_{*j*} 到数据聚合器 DA_{*n*} 之间的瞬时链路增益 $g_{j,n}$ 。又定义二进制变量 $b_{i,k}$ 、 $b_{j,k}$ 描述资源块 RB_{*k*} 与 S_1 、 S_2 中 MTCD 之间的关系。当 $b_{i,k}=1$ 、 $b_{j,k}=1$ 分别表明资源块 RB_{*k*} 被 MTCD_{*i*}、MTCD_{*j*} 所占用, 反之 $b_{i,k}=0$ 、 $b_{j,k}=0$ 。所以在数据聚合器 DA_{*n*} 处, S_1 中 MTCD_{*i*} 在 RB_{*k*} 上的信干噪比为:

$$\text{SINR}_i^k = \frac{p_i^k g_{i,n}}{\sigma^2 + \sum_{i \in S_1 \cap i \neq i} p_i^k g_{i,n} + p_j^k g_{j,n}} \quad (2)$$

其中, p_i^k 、 p_i^k 、 p_j^k 分别为 S_1 中的 MTCD_{*i*}、 S_1 中的 MTCD_{*i*}、 S_2 中的 MTCD_{*j*} 在 RB_{*k*} 上的功率, 分子表示 S_1 中的 MTCD_{*i*} 的信号, 分母第 1、2、3 项分别为零均值的高斯噪声信号、 S_1 中与 MTCD_{*i*} 共用 RB_{*k*} 的 MTCD_{*i*} 所造成的总干扰、 S_2 中具有 QoS 要求的 MTCD_{*j*} 带来的干扰。同理 S_2 中的 MTCD_{*j*} 在资源块 RB_{*k*} 上的信干噪比为:

$$\text{SINR}_j^k = \frac{p_j^k g_{j,n}}{\sigma^2 + \sum_{i \in S_1} p_i^k g_{i,n}} \quad (3)$$

所以针对使得整个 MTC 网络吞吐量最大化的目标, 根据香农信道容量计算公式可以构造出如下最优化问题:

$$\begin{aligned} \max_{b_{i,k}, p_i^k} & \sum_{k=1}^K \left(\sum_{i=1}^{|S_1|} b_{i,k} \ln(1 + \text{SINR}_i^k) + \sum_{j=1}^{|S_2|} b_{j,k} \ln(1 + \text{SINR}_j^k) \right) \\ \text{s.t.} \quad & C_1 \quad \sum_{\forall k \in S_{RB}} b_{j,k} = 1 \\ & C_2 \quad \sum_{\forall k \in S_{RB}} b_{i,k} \geq 1 \\ & C_3 \quad \sum_{\forall k \in S_{RB}} p_i^k \leq p_{\text{MTCD}}^{\max} \\ & C_4 \quad b_{i,k}, b_{j,k} \in \{0, 1\} \end{aligned} \quad (4)$$

其中, C_1 表示 S_2 中每个 MTCD 只允许占用 1 个

RB, C_2 表示 S_1 中每个 MTCD 能与 S_2 中多个 MTCD 进行资源复用, C_3 表示 S_1 中每个 MTCD 具有最大功率限制 $p_{\text{MTCD}}^{\max}$ 。

上述问题属于混合整数非线性规划 (mixed integer nonlinear programming, MINLP) 问题, 通常是 NP 难^[17]的, 很难直接求解, 在本文中使用 Q 学习算法解决。

3 算法的提出

3.1 MTCD 聚簇形成

推荐系统中, 常需要寻找相似度较高的不同用户, 以便将一个用户喜欢的商品推荐给与该用户相似度最高的其他用户。受此启发, 本文提出基于 MTCD 与 DA 之间的余弦相似度进行聚类, 将 MTCD 归属于与其相似度最高, 即 CS 值最大的 DA。由数学知识可知, 若两个向量间的余弦值越接近于 1, 表示两个向量越相似, 反之余弦值越接近于 0, 两个向量越不相似。因此, 基于余弦相似度对 MTCD 进行聚类的, 利用 MTCD 与 DA 之间的信干噪比 SINR 以及距离 d 这两个因子, 分别构造出代表 MTCD 和 DA 的 $h+1$ 维向量。设定向量 A_n 代表 DA_{*n*}, 由于现实中任意 MTCD 和 DA 之间的 SINR 和 d 都是不同的, 所以针对不同的 DA 而言, 代表同一个 MTCD 的向量也是不同的。这里使用向量 $B_{m,n}$ 表示 h 针对 DA_{*n*} 而言, 代表 MTCD_{*m*} 的向量。可以得到 MTCD_{*m*} 与系统中所有 DA 之间的信干噪比大小, 组成一个数组 $\text{SINR}_m = \{\text{SINR}_m^{\text{DA}_1}, \text{SINR}_m^{\text{DA}_2}, \dots, \text{SINR}_m^{\text{DA}_n}, \dots, \text{SINR}_m^{\text{DA}_N}\}$, $\text{SINR}_m^{\text{DA}_n}$ 表示 MTCD_{*m*} 与 DA_{*n*} 之间的信干噪比大小。

下文以 MTCD₁ 和 DA₁ 为例进行说明。向量 A_1 代表 DA₁, 则针对 DA₁ 而言, $h+1$ 维向量 $B_{1,1}$ 代表 MTCD₁, 且 $B_{1,1} = \{B_{1,1}^1, B_{1,1}^2\}$ 。其中, 向量 $B_{1,1}^1 = \{v_{1,0}, v_{1,1}, \dots, v_{1,t}, \dots, v_{1,h-1}\}$ 是一个 h 维向量, $B_{1,1}^2$ 中每个分量的取值依据集合 SINR_{th} 中每个级别的 SINR 而定。MTCD₁ 和 DA₁ 之间的信干噪比表示为 $\text{SINR}_1^{\text{DA}_1}$, 则 $v_{1,t}$ 取值为:

$$v_{1,t} = \begin{cases} 1, & \text{SINR}_1^{\text{DA}_1} \geq \text{SINR}_{\text{th},t} \\ 0, & \text{SINR}_1^{\text{DA}_1} < \text{SINR}_{\text{th},t} \end{cases} \quad (5)$$

$B_{1,1}^2 = \{d_{1,1}\}$ 是一维向量, $d_{1,1}$ 取值依据 MTCD₁ 和 DA₁ 之间的真实距离 $d_{1,1}^*$ 与阈值距离 d_{th} 的大小关系而定。 d_{th} 是为了保证 MTCD₁ 和 DA₁ 之间的信干噪比不低于 γ_1 , 其中, γ_1 为 MTCD₁ 的最低信干噪比, $\gamma_1 \in \text{SINR}_{\text{th}}$ 。MTCD₁ 和 DA₁ 之间的信干噪比可通过式 (6) 得出。

$$\text{SINR}_1^{\text{DA}_1} = \frac{p_1 g_{1,1}}{\sigma^2} = \frac{p_1 G \Gamma_{1,1} (d_{1,1}^*)^{-\alpha}}{\sigma^2} \quad (6)$$

其中, p_1 是 MTCD₁ 发射功率。由于 $\text{SINR}_1^{\text{DA}_1} \geq \gamma_1$, 即 $\frac{p_1 G \Gamma_{1,1} (d_{1,1}^*)^{-\alpha}}{\sigma^2} \geq \gamma_1$, 所以可以求解得到 $d_{1,1}^*$ 必须满足式 (7):

$$d_{1,1}^* \leq \sqrt[\alpha]{\frac{p_1 G \Gamma_{1,1}}{\sigma^2 \gamma_1}} = d_{\text{th}} \quad (7)$$

进而得到向量 $B_{1,1}^2$ 中的 $d_{1,1}$ 分量值为:

$$d_{1,1} = \begin{cases} 1, & d_{1,2}^* \leq d_{\text{th}} \\ 0, & d_{1,2}^* > d_{\text{th}} \end{cases} \quad (8)$$

至此, 代表 MTCD₁ 的 $h+1$ 维向量 $B_{1,1}$ 构造完毕, 而代表 DA₁ 的向量 A_1 , 可将其设置为 $h+1$ 维全 1 向量。从本质上来看, 数据聚合器也是一种 MTCD, 只不过 DA 额外具备资源调度、数据聚合与转发等功能, 且 DA 的数据传输目的地是基站 BS。因此, 当按照系统中 MTCD 与 DA 之间的信干噪比和距离这两个因子来确定向量中各分量的取值时, DA 与其自身之间的距离可视为 0, 即必定满足式 (7), 且信干噪比也会满足最低信干噪比要求。所以使用 $h+1$ 维的全 1 向量表示 DA 是合理的。实际上, 本文中所有 DA 所代表的向量都设置为 $h+1$ 维的全 1 向量。

当分别代表 MTCD₁、DA₁ 的向量 $B_{1,1}$ 、 A_1 构造完成后, 利用式 (9) 可计算出两者之间的相似度 $\text{CS}_{1,1}$:

$$\text{CS}_{1,1} = \frac{B_{1,1} \cdot A_1}{|B_{1,1}| \times |A_1|} = \frac{\sum_{r=1}^{m+1} x_r \times y_r}{|B_{1,1}| \times |A_1|} \quad (9)$$

其中, “ $\cdot$ ” 为点乘符号, 表示向量之间的乘积, $|B_{1,1}|$ 、 $|A_1|$ 分别表示 $B_{1,1}$ 、 A_1 的模长, x_r 、 y_r 分别为 $B_{1,1}$ 、 A_1 的分量。

当求得 MTCD₁ 与 DA₁ 间的相似度 $\text{CS}_{1,1}$ 后, 也可求得系统中任意 MTCD 和 DA 间的相似度, 并依据相似度大小进行聚类, 将 MTCD 归属于与其相似度最大的数据聚合器 DA。详见算法 1。

算法 1 基于余弦相似度的 MTCD 聚类算法
初始化:

M 、 K 、 N 、 h 等数量参数, 以及 h 级 SINR_{th}

循环:

for $h=1:M$

创建数组 CS_m , 用于存储 MTCD _{m} 与所有 DA 间的相似度值;

for $n=1:N$

依据 SINR_{th} 确定 MTCD _{m} 的最低信干噪比 γ_m , 计算 MTCD _{m} 与 MTCD _{m} 间的真实距离 $d_{m,n}^*$, 并根据式 (7) 计算阈值距离 d_{th} ;

依据式 (5)、式 (8) 构造代表 MTCD _{m} 和 DA _{n} 的 $h+1$ 维向量 $B_{m,n}$ 、 A_n ;

依据式 (9) 计算 MTCD _{m} 和 DA _{n} 间的余弦相似度 $\text{CS}_{m,n}$ 值, 并存储在 CS_m 中;

end for

对数组 CS_m 进行降序排序找出最大的元素 $\text{CS}_{m,\text{max}}$, 并将 MTCD _{m} 归属于 $\text{CS}_{m,\text{max}}$ 对应的 DA;

end for

3.2 MTC 网络内部资源分配

(1) 基于 team-Q 学习算法的资源分配策略

Q 学习算法是一种无模型强化学习算法, 无须大量数据训练神经网络, 而是通过智能体 agent 与环境进行交互, 获得奖励, 最终学习最优策略。Q 学习可以使用一个元组表示: $\langle S, A, T, R(s, a), Q \rangle$, S 为环境中有限的状态空间,



A 为可供 agent 选择的有限动作空间, T 为 agent 在状态转移中的概率函数, $R(s,a)$ 为奖励函数, 表示 agent 从当前状态选择特定动作后到达下一状态时的环境反馈, Q 学习过程可以以一张表格形式呈现, 这个表格称为 Q 值表, 表示为 Q 。 Q 值表的行代表不同的状态, 列代表可采取的不同动作, Q 值表中每个值的具体含义是: 该状态下采取该动作的长期期望回报。 Q 学习执行过程对应着 Q 值表更新过程, 而 Q 值表的更新规则如式 (10) 所示。

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \rho \left[R(s, a) + \mu \max_a Q_t(s_t, a) - Q_t(s_t, a_t) \right] \quad (10)$$

其中, ρ 、 μ 分别为学习率、折扣因子, 取值范围均为 $[0,1]$ 。折扣因子用来描述未来回报相对于即时回报的重要性, μ 越接近于 0, 意味着 agent 仅考虑即时奖励, μ 越接近于 1, 表示 agent 更看重未来的奖励。同时, 为了避免算法陷入局部最优, 本文采用了 ε -greedy 算法来进行动作的选择, 即以 ε 概率采取一个随机动作, 以 $1-\varepsilon$ 概率依照 $a_{t+1} = \arg \max_{a \in A} Q(s, a)$ 选取动作。

观察式 (4) 问题可知, 其可看作求解 K 个 RB 上吞吐量总和, 所以将其分解为 K 个并行的资源块与功率分配子任务, 并且每个子任务可看作在单个 RB 上进行学习的任务。在 team-Q 学习算法中, 智能体 agent、状态空间 state、动作空间 action、奖励函数 reward 分别表示如下。

agent (A): 由于预先已为 S_2 中的 MTCD_{*j*} 分配资源块和功率, S_1 中的 MTCD_{*i*} 复用 MTCD_{*j*} 的资源块。因此, 将 S_1 中 MTCD_{*i*} 记为 agent_{*i*}。

state (S): 当处于第 t 个 episode 时, MTCD_{*i*} 在 RB_{*k*} 上的状态空间记为 $s_{i,k}^t$, 其中 episode 为 Q 学习算法的一次训练周期。具体地, $s_{i,k}^t$ 取值为:

$$s_{i,k}^t = \begin{cases} 1, & \text{SINR}_j^t \geq \gamma_j \\ 0, & \text{SINR}_j^t < \gamma_j \end{cases} \quad (11)$$

SINR_j^t 为集合 S_2 中 MTCD_{*j*} 在资源块 RB_{*k*} 上

的瞬时信干噪比, γ_j 为 MTCD_{*j*} 的最低信干噪比要求, $\gamma_j \in \text{SINR}_{th}$, 确保 MTCD_{*j*} 的 QoS 要求得到满足。假定 MTCD_{*j*} 能够向数据聚合器 DA 报告自身的信干噪比值, 而 MTCD_{*i*} 可以从 DA 处获得 MTCD_{*j*} 信息。

action (A): 第 t 个 episode 时, 动作空间为 $A_t(p, f)$, 其中 $p = \{p_1, p_2, \dots, p_l\}$, 表示 MTCD_{*i*} 发射功率由不同的功率级别组成, $f = \{f_1, f_2, \dots, f_K\}$ 为所有资源块集合构成。

reward ($R(s, a)$): 对于 MTCD_{*i*} 而言, 当 MTCD_{*i*} 选择与 MTCD_{*j*} 共用 RB_{*k*} 时, 奖励函数表示为在 RB_{*k*} 上的吞吐量:

$$\text{reward}_{\text{team-Q}}^k = \text{lb}(1 + \text{SINR}_j^k) + \text{lb}(1 + \text{SINR}_i^k) + \sum_{i' \in \lambda_k \cap i' \neq i} \text{lb}(1 + \text{SINR}_{i'}^k) \quad (12)$$

式 (12) 表示在整个 RB_{*k*} 上的吞吐量, 其中, λ_k 表示 S_1 中使用 RB_{*k*} 的 MTCD 集合, 第 1、2、3 项分别为 S_2 中 MTCD_{*j*} 的吞吐量、 S_1 中 MTCD_{*i*} 的吞吐量、 S_1 中除去 MTCD_{*i*} 之外其他与 MTCD_{*j*} 共用 RB_{*k*} 的 MTCD_{*i'*} 产生的吞吐量。在 Q 学习算法中, 奖励函数反映了学习目标, 而由式 (4) 可知本文以系统吞吐量最大化为目标, 这与式 (12) 中奖励函数的定义相契合。

由奖励函数式 (12) 可知, 它不仅需要 MTCD_{*i*} 的有关信息, 还需要 ζ_k 中所有 MTCD 的信息, 所以 team-Q 学习其实是一种集中式算法, 能够确保选择最优的联合动作, 进而可以获得全局最优 Q 值解。然而, 在 team-Q 学习算法中, 由于 agent 的学习目标相同, 均是通过调整自身的功率使得整个 RB_{*k*} 的容量最大, 因此中心化的控制器仅需维护一张 Q 值表, 所有的 agent 都会通过学习更新该 Q 值表。team-Q 学习算法中 Q 值表的空间维度为 $|S| \times |A|^{|\lambda_k|}$, 其中 $|\lambda_k|$ 表示 λ_k 中 MTCD 数量, 这直接导致了 team-Q 学习算法的复杂度随着集合 S_1 中 MTCD 数量的增加而呈现指数级别的增长。

为了解决 team-Q 学习算法复杂度过高问题,提出了一种分布式 Q 学习算法。

(2) 基于 dis-Q 学习算法的资源分配策略

为了解决 team-Q 算法中 Q 值表维度过大导致复杂度过高的问题,提出了 dis-Q 学习算法。在 dis-Q 算法中,不再只有一个 Q 值表,而是维护 $|S|$ 个 Q 值表,每个 Q 值表维度下降为 $|S| \times |A| \times |\lambda_k|$ 。在学习过程中,只有当下一个 Q 值大于当前的 Q 值时,才会在每个 Q 值表中更新该 Q 值。在 dis-Q 学习算法中,智能体 agent、状态空间 state、动作空间 action 的定义皆与 team-Q 学习算法相同,而对于奖励函数则给出了不同于式 (12) 中的定义,详见式 (13)。

$$\text{reward}_{\text{dis-Q}}^{i,k} = \text{lb}(1 + \text{SINR}_i^k) \quad (13)$$

对比式 (12)、式 (13) 可知, $\text{reward}_{\text{team-Q}}^k$ 针对的是整个资源块上全部 MTCD_i 通信链路的吞吐量, $\text{reward}_{\text{dis-Q}}^{i,k}$ 则聚焦单条 MTCD_i 通信链路的吞吐量。网络中各条链路之间需要进行信息的交互,由于在获取奖励值时, team-Q 算法需要获取多个 MTCD 的信息,而 dis-Q 算法则仅需要单个 MTCD 信息即可。因此,在网络的信令消耗方面, dis-Q 学习算法相对于 team-Q 学习算法要小得多。同时,从奖励函数的定义式中也可知, team-Q 学习算法在奖励函数的计算耗时、信息交互的链路条数等均要多于 dis-Q 学习算法,进而使得 dis-Q 学习算法的收敛速度快于 team-Q 算法。然而,从整个系统的吞吐量性能来看, team-Q 算法会略优于 dis-Q 算法。这是由于 team-Q 算法中奖励函数的定义虽然导致信令消耗增加、收敛速度变慢,但是它却保证了整个资源块上所有 MTCD 的总吞吐量,而 dis-Q 算法却仅仅保证的是单条 MTCD 链路的吞吐量。

算法 2 dis-Q 学习资源分配算法

初始化:

ρ, μ , Q 值表, 最大训练周期数 T , 动作空

间 $A_i(p, f)$;

迭代:

随机选择一个初始的状态 $s_0 \in S$;

for $t = 1:T$

产生一个随机数 $\theta \in (0, 1)$;

if $\theta < \varepsilon$

从动作空间 A 中随机采取一个动作

a_{t+1} ;

else

从动作空间 A 中选择下一个 a_{t+1} 使得 Q 值最大: $a_{t+1} = \arg \max_{a \in A} Q(s, a)$;

end if

执行选择的动作 a_{t+1} 后到达下一个状态 s_{t+1} , 由式 (13) 可计算从环境中获得奖励 $\text{reward}_{\text{dis-Q}}^{i,k}$;

根据式 (10) 更新 Q 值表;

end for

4 仿真结果与性能分析

本节主要对本文所提算法的性能进行分析验证,包括收敛性、复杂度和系统吞吐量等,仿真平台是 MATLAB 工具,仿真参数见表 1^[12,20]。

表 1 仿真参数

参数	取值
仿真场景	单蜂窝圆形小区
小区半径	500 m
$P_{\text{MTCD}}^{\text{max}}$	23 dBm
噪声功率谱密度	-174 dBm/Hz
路损因子 α	3
路损常数 G	10^2
慢衰落模型	标准偏差为 8 dB 的对数正态分布

首先对比两种 Q 学习算法的收敛速度。如图 2 所示,可以得到 team-Q 算法和 dis-Q 算法随着迭代次数增加都趋向于收敛,但从迭代次数角度出发, dis-Q 学习算法的收敛速度相比 team-Q 学习



算法提高了近 40%。这是由于在 team-Q 学习算法中, Q 值表的维度远大于 dis-Q 学习算法, 当动作空间和智能体 agent 数量都增大时, team-Q 算法复杂度会呈现指数级增长, 最终导致 dis-Q 学习算法的收敛速度快于 team-Q 学习算法。

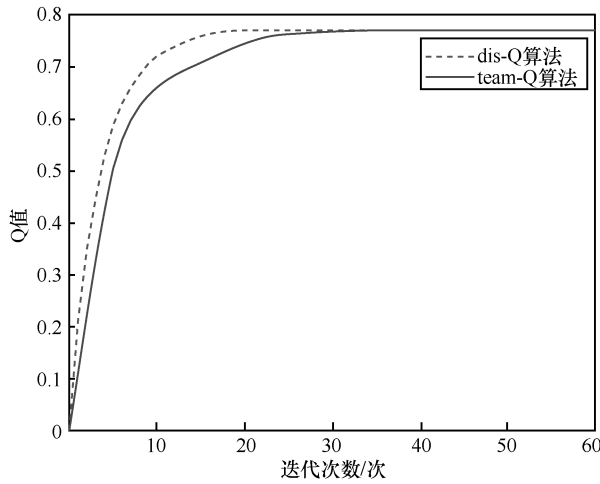


图2 两种 Q 学习算法收敛性分析

上文分析了 team-Q 算法、dis-Q 算法中 Q 值表维度, 分别为 $|S| \times |A|^{\lambda_k}$ 、 $|S| \times |A| \times \lambda_k$ 。图 3 则展示了当 $|\lambda_k|$ 分别取值 1、2、3 时, team-Q 算法和 dis-Q 算法中 Q 值表的维度对比情况。由图 3 可知, 当 $|\lambda_k|=1$ 时, 两者复杂度相当; 当 $|\lambda_k|=2$ 时, dis-Q 算法复杂度为 team-Q 算法的 25%; 当 $|\lambda_k|=3$ 时, dis-Q 的复杂度仅为 team-Q 算法的 4.7%。因此当 $|\lambda_k|$ 取值变大时两者的维度差距也明显变大。

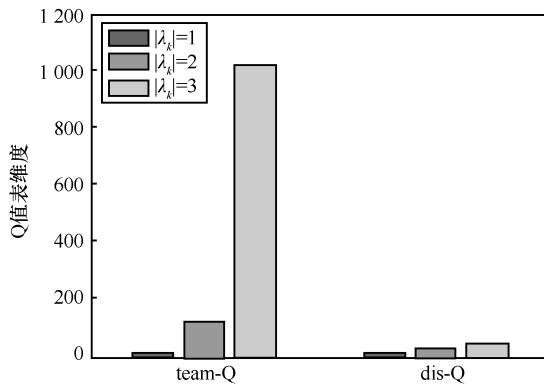


图3 不同 $|\lambda_k|$ 下 team-Q、dis-Q 算法中 Q 值表维度对比

为了体现本文算法的优越性, 将本文提出的 team-Q 和 dis-Q 算法与文献[13]中的动态资源分配算法、贪婪分配算法、随机分配算法的吞吐量进行比较, 结果如图 4 所示。可知 team-Q、dis-Q 算法的吞吐量性能相较于动态资源算法分别平均提高了 16.5%、15.1%, 而传统贪婪算法分别平均提高了 22.3%、21.1%。在 team-Q 算法中 $\text{reward}_{\text{team-Q}}^k$ 代表资源块 RB_k 上所有 MTCD 的吞吐量, 而在 dis-Q 算法中 $\text{reward}_{\text{dis-Q}}^{i,k}$ 仅代表 MTCD_i 单条链路吞吐量。奖励函数的定义不同使得在 Q 表更新过程中, team-Q 算法能够确保多个 MTCD 共用同个 RB 后的整体吞吐量最优, 多个 MTCD 共用同个 RB 后, 会导致多条通信链路间存在干扰, 而 $\text{reward}_{\text{dis-Q}}^{i,k}$ 仅能保证单条通信吞吐量最优。因此, team-Q 学习算法在系统吞吐量方面的表现略高于 dis-Q 算法。然而, 当 MTCD 请求数多于资源数时, 文献[13]的动态资源分配算法并未考虑资源的复用情况, 只是通过优先级拒绝多余的连接请求, 使得系统整体吞吐量下降。

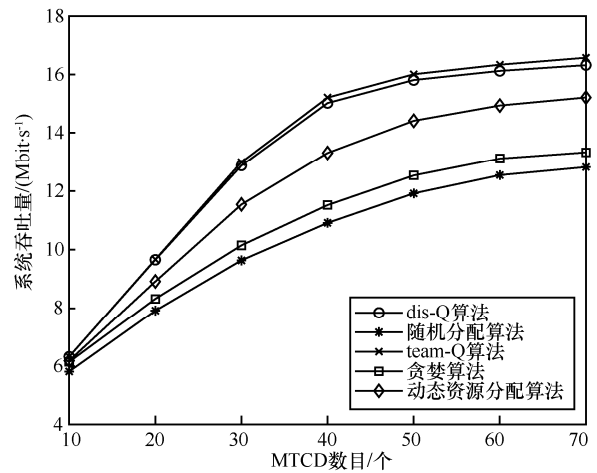


图4 不同算法下系统吞吐量对比

如图 5 所示, 分别对本文提出的基于余弦相似度 CS 的 MTCD 聚类算法、传统 k-means 算法、不考虑任何分组算法的 3 种情况下, 对 team-Q 学习算法、dis-Q 学习算法进行比较。结果表明, 当对 MTCD 进行分组时, 无论是 team-Q 学习算法

下还是 dis-Q 学习算法, 基于余弦相似度的聚类算法的吞吐量比 k -means 算法要高 20.9%、21.4%, 而比未对 MTCD 进行分组时的吞吐量要高 33.1%、35.8%。这是由于 k -means 算法利用欧氏距离进行聚类, 在聚类时只考虑了 MTCD 的地理位置, 然而在系统中为 MTCD 设置了 h 个级别的 QoS 要求, 所以仅依赖 MTCD 的地理位置是无法充分发掘 MTCD 之间的关联性, 从而进行合理聚类。本文提出的基于余弦相似度算法中, 不仅考虑了 MTCD 与 DA 之间的相对地理位置, 还充分利用了 h 级的 QoS 要求, 可以将 MTCD 归类于相似度最高的 DA, 因此可以更好地协调聚簇内的干扰, 提高系统的吞吐量。若不对 MTCD 进行分组, 会造成瞬时直接接入基站的 MTCD 连接数变大, 使得基站接入负担增大, 导致部分 MTCD 接入失败, 进而使得系统吞吐量严重下降。

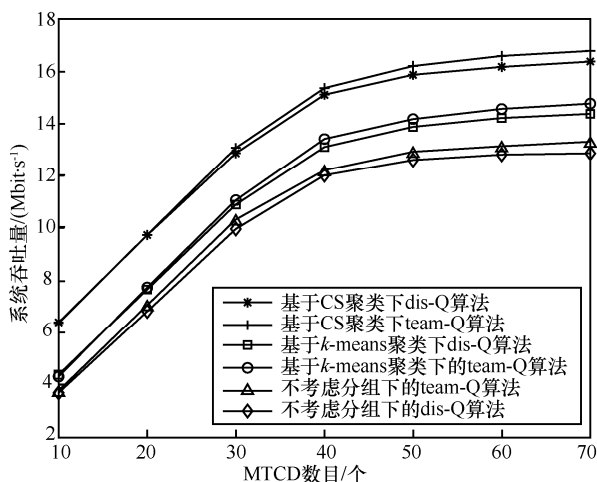


图5 不同聚类算法下系统吞吐量对比

5 结束语

本文研究了在 mMTC 场景中以系统吞吐量最优化为目标的资源分配问题。首先, 提出了一种基于余弦相似度的聚类算法, 根据 MTCD 与 DA 之间的相对位置和 QoS 要求, 将 MTCD 分组。该算法能充分发掘出 MTCD 之间的关联性, 能更好地协调 MTC 聚簇内的干扰, 有利于提升系统性

能。此外, 针对 MTC 网络中的资源分配问题, 提出了 team-Q 学习算法和 dis-Q 学习算法。仿真结果表明, 两种 Q 学习算法对系统吞吐量的提升作用相较于对比算法均有较大幅度的提高, 其中 team-Q 算法在系统吞吐量性能上略优于 dis-Q 算法, 但是 dis-Q 算法在信令消耗、收敛速度方面明显优于 team-Q 算法, 这也更加符合“绿色通信”的理念。

参考文献:

- [1] CHEN S Y, MA R F, CHEN H H, et al. Machine-to-machine communications in ultra-dense networks—A survey[J]. IEEE Communications Surveys & Tutorials, 2017, 19(3): 1478-1503.
- [2] 钱志鸿, 王义君. 物联网技术与应用研究[J]. 电子学报, 2012, 40(5): 1023-1029.
QIAN Z H, WANG Y J. IoT technology and application[J]. Acta Electronica Sinica, 2012, 40(5): 1023-1029.
- [3] Service-aware transport network: opportunities and challenges[J]. Proceedings of SPIE - The International Society for Optical Engineering, 2005.
- [4] ZHOU Y Q, TIAN L, LIU L, et al. Fog computing enabled future mobile communication networks: a convergence of communication and computing[J]. IEEE Communications Magazine, 2019, 57(5): 20-27.
- [5] Cisco visual networking index: global mobile data traffic forecast update 2014-2019[EB]. 2014.
- [6] LIANG L, XU L, CAO B, et al. A cluster-based congestion-mitigating access scheme for massive M2M communications in internet of things[J]. IEEE Internet of Things Journal, 2018, 5(3): 2200-2211.
- [7] GHAVIMI F, LU Y W, CHEN H H. Uplink scheduling and power allocation for M2M communications in SC-FDMA-based LTE-A networks with QoS guarantees[J]. IEEE Transactions on Vehicular Technology, 2017, 66(7): 6160-6170.
- [8] GAO H, XU X D, HAN S J. Homogeneous clustering algorithm based on average residual energy for energy-efficient MTC networks[C]//Proceedings of 2018 24th Asia-Pacific Conference on Communications (APCC). Piscataway: IEEE Press, 2018: 28-33.
- [9] HUSSAIN F, HUSSAIN R, ANPALAGAN A, et al. A new block-based reinforcement learning approach for distributed resource allocation in clustered IoT networks[J]. IEEE Transactions on Vehicular Technology, 2020, 69(3): 2891-2904.
- [10] XU Y Q, FENG G, LIANG L, et al. MTC data aggregation for 5G network slicing[C]//Proceedings of 2017 23rd Asia-Pacific

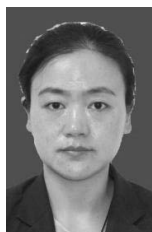


- Conference on Communications (APCC). Piscataway: IEEE Press, 2017: 1-6.
- [11] 王鑫, 邱玲. H2H 与 M2M 共存场景的准入控制及资源分配[J]. 中国科学院大学学报, 2016, 33(3): 427-432.
- WANG X, QIU L. Admission control and resource allocation of H2H & M2M co-existence scenario[J]. Journal of University of Chinese Academy of Sciences, 2016, 33(3): 427-432.
- [12] 蒋继胜, 朱晓荣. H2H 与 M2M 共存场景下的上行资源分配算法[J]. 电子学报, 2018, 46(5): 1259-1264.
- JIANG J S, ZHU X R. An uplink resource allocation algorithm under the scenario of coexistence of H2H & M2M based on knapsack model[J]. Acta Electronica Sinica, 2018, 46(5): 1259-1264.
- [13] SALAM T, REHMAN W U, TAO X F. Cooperative data aggregation and dynamic resource allocation for massive machine type communication[J]. IEEE Access, 2018, 6: 4145-4158.
- [14] 郭涛, 李有明, 雷鹏, 等. MIMO 中继系统中一种基于用户 QoS 的资源分配方法[J]. 电信科学, 2015, 31(4): 121-126.
- GUO T, LI Y M, LEI P, et al. A resource allocation scheme based on user's QoS in MIMO relay system[J]. Telecommunications Science, 2015, 31(4): 121-126.
- [15] 张海波, 向煜, 刘开健, 等. 基于 D2D 通信的 V2X 资源分配方案[J]. 北京邮电大学学报, 2017, 40(5): 92-97.
- ZHANG H B, XIANG Y, LIU K J, et al. V2X resource allocation scheme based on D2D communication[J]. Journal of Beijing University of Posts and Telecommunications, 2017, 40(5): 92-97.
- [16] 刘佳言, 秦鹏, 赵雄文, 等. 基于容量最大化的 mMTC 场景的资源分配问题研究[J]. 电力信息与通信技术, 2020, 18(12): 17-22.
- LIU J Y, QIN P, ZHAO X W, et al. Research on resource allocation of mMTC scenario based on capacity maximization[J]. Electric Power Information and Communication Technology, 2020, 18(12): 17-22.
- [17] SHARMA S K, WANG X B. Toward massive machine type communications in ultra-dense cellular IoT networks: current issues and machine learning-assisted solutions[J]. IEEE Communications Surveys & Tutorials, 2020, 22(1): 426-471.
- [18] HUSSAIN F, HASSAN S A, HUSSAIN R, et al. Machine learning for resource management in cellular and IoT networks: potentials, current solutions, and open challenges[J]. IEEE Communications Surveys & Tutorials, 2020, 22(2): 1251-1275.
- [19] NIKOPOUR H, BALIGH H. Sparse code multiple access[C]//Proceedings of 2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC). Piscataway: IEEE Press, 2013: 332-336.
- [20] KAI C H, LI H, XU L, et al. Joint subcarrier assignment with power allocation for sum rate maximization of D2D communications in wireless cellular networks[J]. IEEE Transactions on Vehicular Technology, 2019, 68(5): 4748-4759.

[作者简介]



余云河 (1995-), 男, 南京邮电大学通信与信息工程学院硕士生, 主要研究方向为大规模机器类通信网络中的资源分配。



孙君 (1980-), 女, 南京邮电大学硕士生导师, 主要研究方向为无线网络资源管理。