

大规模分布式天线系统半动态双向分簇算法^①

唐 伟^{②*} 孙 强^{③***} 高 翔^{***} 王 珏^{*} 李宇晖^{*} 赵欣钰^{*}

(* 南通大学电子信息学院 南通 226019)

(** 南通先进通信技术研究院 南通 226019)

(*** 国网江苏省电力公司常州供电公司 常州 213000)

摘 要 针对现有分簇算法大多假设在理想条件下利用瞬时信道状态信息(CSI),但在回程链路容量受限的情况下,难以共享所有用户的瞬时 CSI 的问题,提出了一种半动态双向选择分簇算法。该算法利用长时 CSI 进行分簇,减轻了回程链路的负担;且该算法考虑了远程天线单元间的双向选择,克服了传统算法只考虑单向选择导致的系统性能受限的问题;此外,该算法通过限制协作簇规模并采用贪婪算法进行协作簇的合并,降低了计算复杂度。仿真结果表明,所提算法的频谱效率远高于静态分簇算法并且接近利用瞬时 CSI 的动态分簇算法。

关键词 大规模分布式天线系统,分簇算法,信道状态信息(CSI),频谱效率

0 引 言

大规模多输入多输出(multiple-input multiple-output, MIMO)是一种新的物理层技术^[1],可利用基站端大幅提高的天线数目进一步挖掘空间资源,提升系统的频谱效率和功率效率,是 5G 中的关键技术之一^[2]。宽带无线移动通信系统中使用大规模 MIMO 可以分为两类:大规模集中式天线系统和大规模分布式天线系统(large-scale distributed antenna system, L-DAS)。在 L-DAS 中,远程天线单元(remote antenna unit, RAU)密集分布在整个区域中,每个 RAU 通过高速回程链路与中央处理单元(central processing unit, CPU)相连接。和传统的分布式天线系统不同, L-DAS 可以更进一步缩短用户和 RAU 之间的距离。然而, L-DAS 会带来更多的 RAU 间干扰,特别是处于边缘地区的 RAU。一种理想的方法是所有的 RAU 都协作,这样可以获得最优的性能,

但这需要获取并共享所有用户的信道状态信息(channel state information, CSI),在回程链路容量受限的情况下,这是难以实现的。为了解决这个问题,可以通过 RAU 分簇来减少协作的规模。

分簇算法已经被广泛应用于基站协作中,其主要分为两种:静态分簇和动态分簇。静态分簇算法复杂度较低,使用较少的信令开销,但只考虑基站地理位置,不考虑信道特性。文献[3]提出了一种基于静态分簇的 L-DAS 用户调度算法,有效提高了系统和速率,但静态分簇对抑制簇间干扰具有一定的局限性。动态分簇算法周期性地根据信道特性对小区进行分簇^[4]。Papadogiannis 等人^[5]首先提出了一种贪婪动态分簇算法,该算法使用瞬时 CSI 贪婪的搜索协作簇。文献[6]提出了一种基于信噪比的动态分簇算法,并且通过设置门限值来降低算法的复杂度,但作者只考虑了最大的信噪比而忽略了 RAU 之间协作的双向性。在文献[4]中,作者提出了一种基于最大协作增益的动态分簇算法,改善了小区

① 国家自然科学基金(61501264)和南通大学-南通智能信息技术联合研究中心(KFKT2017B01)资助项目。

② 男,1997 年生;研究方向:大规模分布式天线系统;E-mail:1511011038@dzxy.ntu.edu.cn

③ 通讯作者,E-mail: sunqiang@ntu.edu.cn

(收稿日期:2018-04-09)

边缘用户的传输速率,但同样忽略了协作的双向性。综上所述,传统的动态分簇算法研究普遍存在以下问题:

(1) 依赖瞬时 CSI,获取并共享所有用户的瞬时 CSI,将会带来巨大的信令开销和回程链路开销,而在 L-DAS 中,由于回程链路容量的限制,这是难以实现的。

(2) 单向性 RAU 选择,每个 RAU 自私地选择可以给它带来最大信道增益的协作簇,破坏了最佳的分簇结构。

针对以上两个问题,本文提出了一种半动态双向选择分簇算法。该算法利用长时 CSI 对 RAU 进行分簇,可以大幅减轻回程链路负担并且获得较高的系统和速率。同时,该算法考虑了 RAU 协作的双向性,能获得近似最优的分簇结果。

1 系统模型

1.1 网络模型

假设 L-DAS 网络模型是一个 2 维 $N \times N$ 的正六边形拓扑结构,如图 1 所示。每个正六边形都有相应的坐标 (i, j) ,其中心圆圈代表 RAU,每个 RAU 的天线数为 N_t ,天线高度为 h 。RAU 到其最近的正六边形顶角的距离为 R ,则任意相邻的两个 RAU 之间距离为 $\sqrt{3}R$ 。模型的覆盖范围是边长为 1km 的

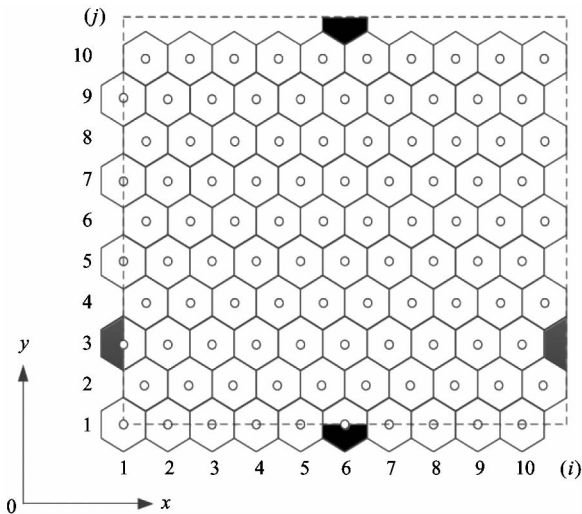


图 1 2 维正六边形拓扑结构 ($N=10$)

正方形。用户随机分布在系统模型中,每个用户配有单根天线。为了避免边界效应,假设该拓扑结构为一个圆环,即图 1 中矩形区域的相对侧实际上是确定的。例如,正六边形 $(1,3)$ 处于正方形区域外的阴影部分,又出现在了它的对面一侧。

1.2 信道模型

信号从发射端经过无线信道到达接收端将受到大尺度衰落和小尺度衰落的影响,本文的信道模型考虑了这两种衰落带来的影响。RAU m 到用户 k 的下行信道可以表示为

$$\mathbf{h}_{k,m} = \sqrt{\beta_{k,m}} \mathbf{g}_{k,m} \tag{1}$$

其中, $\mathbf{g}_{k,m}$ 为用户 k 到 RAU m 的小尺度衰落,其各元素是 0 均值、单位方差的独立同分布复高斯随机变量。 $\beta_{k,m}$ 为用户 k 到 RAU m 的大尺度衰落,其表达式为

$$\beta_{k,m} = f_{k,m} d_{k,m}^{-\eta} \tag{2}$$

其中, $f_{k,m}$ 是从 RAU m 到用户 k 的阴影衰落变量,服从方差为 σ^2 的对数正态分布; η 是路径损耗因子, $d_{k,m}$ 是从 RAU m 到用户 k 的距离。

1.3 信号模型

在系统模型中,RAU 集合为 $\mathcal{M} = \{1, 2, \dots, M\}$,服务的用户集合为 $\mathcal{K} = \{1, 2, \dots, K\}$,所有用户都是随机分布在网络模型中,并且每个用户归属于大尺度衰落最大的 RAU。几个 RAU 一起协作称为协作簇,定义协作簇集合为 $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_L\}$, L 为协作簇集合的个数。定义协作簇 $\mathcal{C}_i$ 中的用户集合为 $\mathcal{U}_i$,其中, $\mathcal{U}_i = \{i_1, i_2, \dots, i_k\}$,那么,用户集合为 $\mathcal{U} = \{\mathcal{U}_1, \mathcal{U}_2, \dots, \mathcal{U}_L\}$ 。

在协作簇 $\mathcal{C}_i$ 中,用户 k 的接收信号矢量为

$$\begin{aligned} \mathbf{y}_k^{(\mathcal{C}_i)} = & \underbrace{\mathbf{h}_k^{(\mathcal{C}_i)} \mathbf{w}_k^{(\mathcal{C}_i)} s_k}_{\text{有用信号}} + \underbrace{\sum_{m \in \mathcal{U}_i \setminus k} \mathbf{h}_k^{(\mathcal{C}_i)} \mathbf{w}_m^{(\mathcal{C}_i)} s_m}_{\text{簇内干扰}} \\ & + \underbrace{\sum_{j \neq i} \sum_{b \in \mathcal{U}_j} \mathbf{h}_k^{(\mathcal{C}_i)} \mathbf{w}_b^{(\mathcal{C}_j)} s_b}_{\text{簇间干扰}} + \underbrace{\mathbf{z}_k}_{\text{干扰}} \end{aligned} \tag{3}$$

其中, $\mathbf{h}_k^{(\mathcal{C}_i)}$ 是协作簇 $\mathcal{C}_i$ 中所有 RAU 到用户 k 的信道矢量, $\mathbf{h}_k^{(\mathcal{C}_j)}$ 是协作簇 $\mathcal{C}_j$ 中所有 RAU 到用户 k 的信道矢量, $\mathbf{w}_k^{(\mathcal{C}_i)}$ 是协作簇 $\mathcal{C}_i$ 对用户 k 的预编码矢量, $\mathbf{w}_m^{(\mathcal{C}_i)}$ 是协作簇 $\mathcal{C}_i$ 对用户 m 的预编码矢量, $\mathbf{w}_b^{(\mathcal{C}_j)}$ 是协作簇 $\mathcal{C}_j$ 对用户 b 的预编码矢量, s_k 是协作簇 $\mathcal{C}_i$

中所有 RAU 对于用户 k 的传输信号, s_m 是协作簇 $\mathcal{V}_i$ 中所有 RAU 对于用户 m 的传输信号, s_b 是协作簇 V_i 中所有 RAU 对于用户 b 的传输信号, z_k 是 0 均值和单位方差的高斯白噪声。

簇内干扰可以通过预编码矩阵来消除,如果协作簇内只有一个用户,则采用最大比传输预编码^[7],最大比传输预编码的表达式为

$$\mathbf{w}_k^{(\mathcal{V}_i)} = \frac{(\mathbf{h}_k^{(\mathcal{V}_i)})^H}{\|\mathbf{h}_k^{(\mathcal{V}_i)}\|} \quad (4)$$

如果协作簇内有多个用户,则采用迫零预编码^[8]来完全消除簇内干扰,迫零预编码的表达式为

$$\mathbf{w}_k^{(\mathcal{V}_i)} = \|\mathbf{h}_k^{(\mathcal{V}_i)}\|^H [\mathbf{h}_k^{(\mathcal{V}_i)} (\mathbf{h}_k^{(\mathcal{V}_i)})^H]^{-1} \quad (5)$$

为了简单起见,假设所有 RAU 的发射功率 p 是相同的,因此,用户 k 的信干噪比(signal to interference plus noise ratio, SINR)为

$$\gamma_k^{(\mathcal{V}_i)} = \frac{p \|\mathbf{h}_k^{(\mathcal{V}_i)} \mathbf{w}_k^{(\mathcal{V}_i)}\|^2}{p \sum_{j \neq i, j \in \mathcal{V}_j} \|\mathbf{h}_k^{(\mathcal{V}_j)} \mathbf{w}_b^{(\mathcal{V}_j)}\|^2 + 1} \quad (6)$$

由此,L-DAS 的系统速率为

$$R = \sum_{i=1}^L \sum_{k \in \mathcal{V}_i} \log_2(1 + \gamma_k^{(\mathcal{V}_i)}) \quad (7)$$

2 分簇算法

现有动态分簇算法普遍采用单向 RAU 选择,虽然每个 RAU 可以选择带来最大信道增益的协作簇,

但是却破坏了最佳的分簇结构。另一方面,其大多利用瞬时 CSI,但是在实际情况中,瞬时 CSI 难以获取,并且增加了回程链路负担。一般而言,大尺度衰落(长时 CSI)至少在 40 个小尺度衰落相干时间的间隔内保持不变^[9]。考虑到长时 CSI 容易获取,在一定时间内相对于瞬时 CSI 是保持不变的。由此,本文利用长时 CSI 对 RAU 进行分簇。

2.1 双向协作

如图 2(a)和图 2(b)所示,协作区域内有 2 个 RAU 和 1 个用户。实线代表有用信号,虚线代表干扰信号。对于 RAU A 是否想和 RAU B 协作取决于从 RAU B 到用户 k 的干扰。

在图 2(a)中,用户 k 不位于 RAU B 的边缘附近,从 RAU B 到用户 k 的干扰比较小,假如 RAU A 和 RAU B 协作,只会给用户 k 带来很小的信道增益。在图 2(b)中,用户 k 位于 RAU B 的边缘附近,从 RAU B 到用户 k 的干扰比较大,假如 RAU A 和 RAU B 协作,将会给用户 k 带来较大的信道增益。因此,用户 k 的 SINR 可衡量 RAU A 对 RAU B 的单向协作意愿。当 RAU B 对应的区域内存在另一个用户 b 时,如图 3 所示,假设 RAU A 归属于协作簇 $\mathcal{V}_i$, RAU B 归属于协作簇 $\mathcal{V}_j$,此时再次讨论 RAU A 与 RAU B 之间的协作意愿,即协作簇 $\mathcal{V}_i$ 与协作簇 $\mathcal{V}_j$ 的协作意愿。定义 $\gamma_k^{(\mathcal{V}_i)}$ 为协作簇 $\mathcal{V}_i$ 中用户 k 的 SINR, $\gamma_b^{(\mathcal{V}_j)}$ 为协作簇 $\mathcal{V}_j$ 中用户 b 的 SINR。由于簇

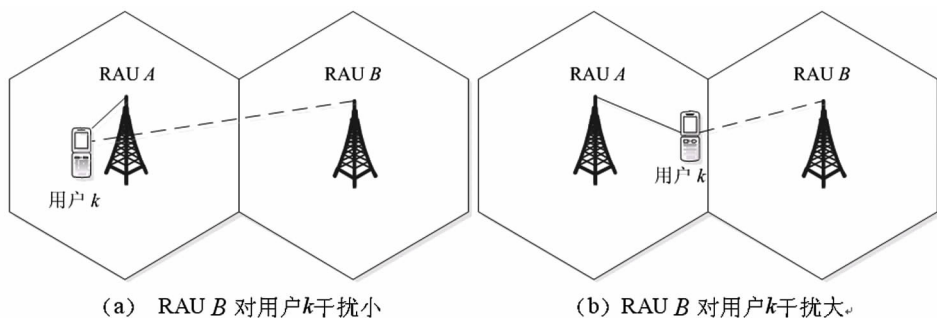


图 2 RAU 对用户的干扰大小

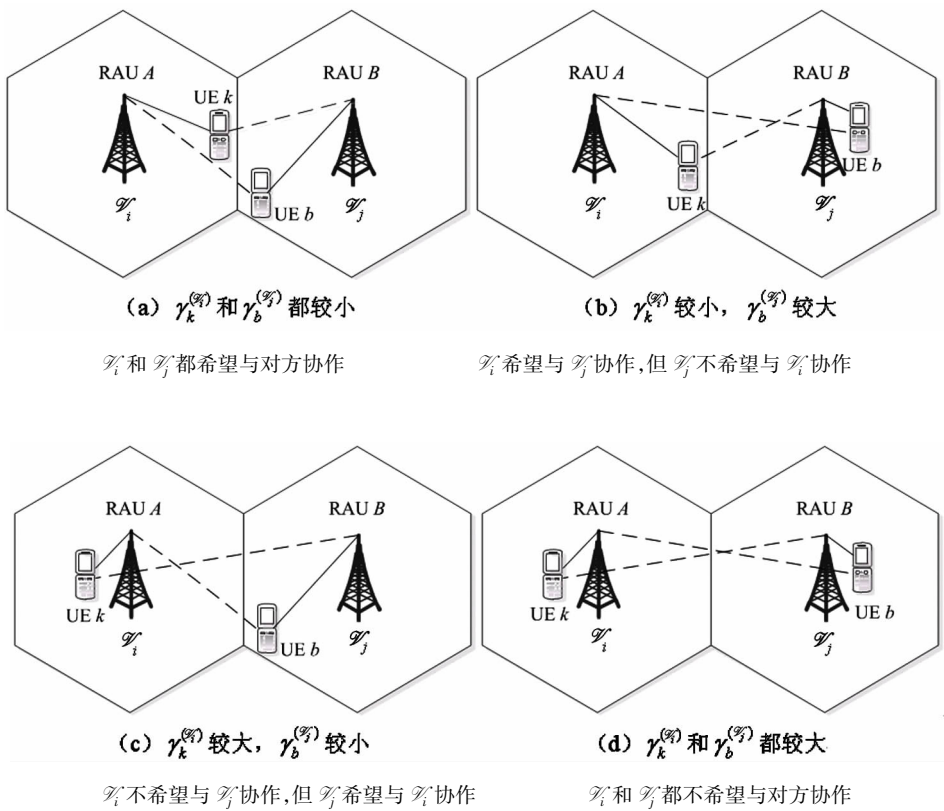


图 3 4 种不同协作场景

内干扰可以通过迫零预编码来消除,可以得到 $\gamma_k^{(\mathcal{U}_i)}$ 和 $\gamma_b^{(\mathcal{U}_j)}$ 的表达式

$$\gamma_k^{(\mathcal{U}_i)} = \frac{p \| \mathbf{h}_k^{(\mathcal{U}_i)} \mathbf{w}_k^{(\mathcal{U}_i)} \|^2}{p \left(\sum_{b \in \mathcal{U}_j} \| \mathbf{h}_k^{(\mathcal{U}_i)} \mathbf{w}_b^{(\mathcal{U}_j)} \|^2 + \sum_{r \neq i, j, c \in \mathcal{U}_r} \| \mathbf{h}_k^{(\mathcal{U}_i)} \mathbf{w}_c^{(\mathcal{U}_r)} \|^2 \right) + 1} \quad (8)$$

$$\gamma_b^{(\mathcal{U}_j)} = \frac{p \| \mathbf{h}_b^{(\mathcal{U}_j)} \mathbf{w}_b^{(\mathcal{U}_j)} \|^2}{p \left(\sum_{k \in \mathcal{U}_i} \| \mathbf{h}_b^{(\mathcal{U}_j)} \mathbf{w}_k^{(\mathcal{U}_i)} \|^2 + \sum_{r \neq i, j, c \in \mathcal{U}_r} \| \mathbf{h}_b^{(\mathcal{U}_j)} \mathbf{w}_c^{(\mathcal{U}_r)} \|^2 \right) + 1} \quad (9)$$

协作簇 $\mathcal{U}_i$ 对协作簇 $\mathcal{U}_j$ 的协作意愿跟 $\gamma_k^{(\mathcal{U}_i)}$ 有关, 协作簇 $\mathcal{U}_j$ 对协作簇 $\mathcal{U}_i$ 的协作意愿和 $\gamma_b^{(\mathcal{U}_j)}$ 有关, $\gamma_k^{(\mathcal{U}_i)}$ 或者 $\gamma_b^{(\mathcal{U}_j)}$ 越小, 意味着协作意愿越大。在多数情况下 $\gamma_k^{(\mathcal{U}_i)} \neq \gamma_b^{(\mathcal{U}_j)}$, 也就是说协作簇直接的协作是有方向性的。根据用户位置的不同, 协作簇间的协作意愿可以分为 4 种情况, 如图 3 所示。

当 $\gamma_k^{(\mathcal{U}_i)} \neq \gamma_b^{(\mathcal{U}_j)}$ 时, 协作簇之间的协作意愿可能是相同的, 也可能是不同的。在图 3(b) 和图 3(c)

的情况下, 一方协作意愿很强, 而另一方协作意愿很弱, 此时若只考虑一方的单向协作意愿, 不能得到最佳的分簇结果, 因此, 需要考虑协作簇 $\mathcal{U}_i$ 和协作簇 $\mathcal{U}_j$ 的双向协作意愿。

考虑双向协作意愿的动态分簇过程可以近似为合并双向 SINR 增益的平均值最大的 2 个簇, 该分簇问题可以描述为

$$\left. \begin{aligned} & \max \left(\frac{\sum_{k \in \mathcal{U}_i} \alpha_k^{(\mathcal{U}_i)} + \sum_{b \in \mathcal{U}_j} \alpha_b^{(\mathcal{U}_j)}}{|\mathcal{U}_i| + |\mathcal{U}_j|} \right) \\ & \text{s. t. } \begin{cases} |\mathcal{U}_i| + |\mathcal{U}_j| \leq Q & \forall \mathcal{U}_i \subseteq \mathcal{U} \\ \mathcal{U}_i \cap \mathcal{U}_j = \phi & \forall i, j \end{cases} \end{aligned} \right\} \quad (10)$$

因为用户的和速率和其 SINR 有关, 可以定义 SINR 增益为用户合并后比上用户合并前的 SINR 比值, 以此来近似反应用户合并前后的和速率变化。在式(10)中 $\alpha_k^{(\mathcal{U}_i)}$ 是用户 k 的 SINR 增益, $\alpha_b^{(\mathcal{U}_j)}$ 是用户 b 的 SINR 增益, $|\mathcal{U}_i|$ 为协作簇 $\mathcal{U}_i$ 中用户个数, $|\mathcal{U}_j|$ 为协作簇 $\mathcal{U}_j$ 中用户个数, Q 为一个协作簇中 RAU 最多个数, 即簇规模门限值。

上述问题是一个组合优化问题, 最优解可通过

穷举法求解,但是复杂度很高,在实际应用中不可实施,于是本文采取复杂度较低的贪婪算法获得次优解,其具体步骤见 2.2 节。

2.2 双向选择半动态分簇算法

考虑回程链路容量受限和实现复杂度要求,本文提出了一种利用长时 CSI 的双向选择半动态分簇算法,所提出的算法以最大化系统和速率、减轻回程链路负担和降低算法的复杂度为目的。在所提算法中,还考虑了协作簇的双向选择性和用户数不同所带来的影响。算法的具体步骤如下。

初始化 设置 RAU 集合 $\mathcal{M} = \{1, 2, \dots, M\}$, 用户集合 $\mathcal{K} = \{1, 2, \dots, K\}$, 协作簇集合 $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_L\}$, 每个协作簇中用户集合 $\mathcal{U}_i = \{u_{i1}, u_{i2}, \dots, u_{iL}\}$ 。

步骤 1 利用长时信道信息,每个用户选择大尺度衰落最大的 RAU。

```
for  $k = 1, 2, \dots, K$ 
 $m^* = \arg \max_m \beta_{k,m}$ , for  $m = 1, 2, \dots, M$ 
end for
```

步骤 2 计算任意两个协作簇之间的双向 SINR 增益的平均值。

```
for  $i = 1; i < L; i++$ 
  for  $j = i + 1; j \leq L; j++$ 
    for  $k = 1; k < |\mathcal{U}_i|; k++$ 
       $\alpha_k^{(\mathcal{U}_i)} = \frac{\gamma_k^{(\mathcal{U}_i)}}{\gamma_k^{(\mathcal{U}_j)}}$ 
    end for
    for  $b = 1; b < |\mathcal{U}_j|; b++$ 
       $\alpha_b^{(\mathcal{U}_j)} = \frac{\gamma_b^{(\mathcal{U}_j)}}{\gamma_b^{(\mathcal{U}_i)}}$ 
    end for
     $\overline{\alpha}^{(\mathcal{U}_i)} = \frac{\sum_{k \in \mathcal{U}_i} \alpha_k^{(\mathcal{U}_i)} + \sum_{b \in \mathcal{U}_j} \alpha_b^{(\mathcal{U}_j)}}{|\mathcal{U}_i| + |\mathcal{U}_j|}$ 
  end for
end for
```

步骤 3 协作簇进行合并

```
while  $\exists \overline{\alpha}^{(\mathcal{U}_i)} > 0$ 
   $(i^*, j^*) = \arg \max_{i,j} \overline{\alpha}^{(\mathcal{U}_i)}$ 
   $d = i^*, e = j^*$ 
  if  $|\mathcal{U}_d| + |\mathcal{U}_e| \leq Q$  then
    更新  $\mathcal{U}$  和  $L$ ,  $\mathcal{U}_d = \mathcal{U}_d \cup \mathcal{U}_e$ ,  $\mathcal{U}_e = \emptyset$ ,
     $\mathcal{U}_d = \mathcal{U}_d \cup \mathcal{U}_e$ ,  $\mathcal{U}_e = \emptyset$ ,  $L = L - 1$ ,
    返回步骤 2
  else
    更新  $\overline{\alpha}^{(\mathcal{U}_e)} = 0$ , 返回步骤 3
  end if
end while
```

步骤 3 利用贪婪算法,在限制簇规模的条件下

进行协作簇的合并。协作簇组合数共有 $\sum_{j=2}^M C_j^2$, 因此,所提算法的复杂度为 $O(M^3)$, 而采取穷举算法的复杂度为 $O(M \cdot M!)$ 。

3 仿真结果及分析

本节对上述算法进行仿真,比较各算法对用户数相同与用户数不同情况下系统和速率的影响。仿真场景如图 1 所示,共有 100 个 RAU, 2 个相邻 RAU 之间的间距为 125 m, RAU 发射天线数 $N_t = 2$, 天线高度 $h = 20$ m, 用户接收天线数 $N_r = 1$ 。RAU 边缘接收功率为 -10 dB ~ 10 dB, 噪声功率为 -102 dB, 阴影衰落的方差 $\sigma^2 = 8$ dB, 路径损耗因子 $\eta = 4$ 。簇规模门限值 $Q = 4$, 用户数 K 为 $5 \sim 25$, 用户在图 1 虚线所连的正方形区域内随机撒点, 每个信道由 10 000 个时隙组成。为了防止实验的偶然性,进行了 50 次用户随机分布实验,得到最后的仿真数据。为了更好地比较所提算法,把所提算法和另外 7 种算法进行了比较。

(1) 所有 RAU 都协作: 所有 RAU 共同服务所有的用户。

(2) 利用瞬时 CSI 的双向选择动态分簇算法: 该算法使用了瞬时 CSI, 并且考虑了 2.1 节所提的双向选择。

(3) 利用瞬时 CSI 的单向选择动态分簇算

法^[6]:该算法使用了瞬时 CSI,但只考虑了单向选择。

(4)利用长时 CSI 的双向选择半动态分簇算法:即为所提的算法,使用了长时 CSI,并且考虑了双向选择。

(5)利用长时 CSI 的单向选择半动态分簇算法:该算法使用了长时 CSI,但只考虑了单向选择。

(6)不分簇:每个用户只选择大尺度衰落最大的 RAU,并且遭受着其他 RAU 所带来的干扰。

(7)静态分簇(2 RAUs):2 个 RAU 协作的静态分簇算法^[3]。

(8)静态分簇(4 RAUs):4 个 RAU 协作的静态分簇算法。

图 4 给出了当用户数为 10 的时候,不同算法的系统速率曲线。如果 RAU 都不协作,RAU 间干扰将会很大,导致系统和速率很低。所有 RAU 一起协作的可以实现和速率最大化,但在实际中几乎不可能实现。静态分簇算法可以一定程度上抑制簇间干扰,但由于用户是随机分布在系统模型中的,处在 RAU 边缘的用户仍然会受到严重的簇间干扰。因此,静态分簇算法的系统速率只比 RAU 都不协作的情况有了小幅度的提升。随着 RAU 边缘接收功率的提升,簇间干扰将会变大,长时 CSI 双向选择半动态分簇算法的系统速率要明显优于静态分簇算法。如果不考虑 RAU 之间的双向选择性,系统和速率将比考虑了双向选择性的算法低。尽管利用瞬时 CSI 的双向选择和单向选择动态分簇算法要优于利用长时 CSI 双向选择半动态分簇算法,但它们都会带来更多的反馈开销,加重回程链路容量的负担。当然,所有 RAU 一起协作可以实现和速率最大化,但在实际中几乎不可能实现。

图 5 给出了当 RAU 边缘功率为 10dB 时,系统和速率曲线随用户数变化的曲线。随着用户数的增加,长时 CSI 双向选择半动态分簇算法的系统速率始终优于静态分簇算法。当用户数为 25 时,长时 CSI 双向选择半动态分簇算法的系统速率要比静态分簇(4 RAUs)算法的系统速率高 20% 左右。另一方面,长时 CSI 双向选择半动态分簇算法的系统速率只比瞬时 CSI 双向选择动态分簇算法的系

统和速率低 3% 左右。

表 1 不同分簇算法执行所需时间($K=10$)

分簇算法	时间(s)
瞬时 CSI 双向选择动态分簇	179.4378
瞬时 CSI 单向选择动态分簇	129.3236
长时 CSI 双向选择半动态分簇	0.0405
长时 CSI 单向选择半动态分簇	0.0177
静态分簇(2RAUs)	0.0008
静态分簇(4RAUs)	0.0013

表 1 给出了在 Matlab R2014a 中不同分簇算法执行所需时间。由于利用瞬时 CSI 的动态分簇算法需要每个时隙进行分簇,其执行所需时间远大于长时 CSI 双向选择半动态分簇,因此所提算法的复杂度可以显著降低。

4 结 论

本文主要研究了回程链路容量受限的 L-DAS 的 RAU 分簇算法,提出了一种利用长时 CSI 的双向选择半动态分簇算法。该算法主要是利用长时 CSI,根据协作簇之间的双向 SINR 增益的平均值最大对 RAU 进行分簇。仿真结果表明,该算法的性能要远优于静态分簇算法,接近使用瞬时 CSI 的单向选择动态分簇算法,并且其复杂度可以显著降低。

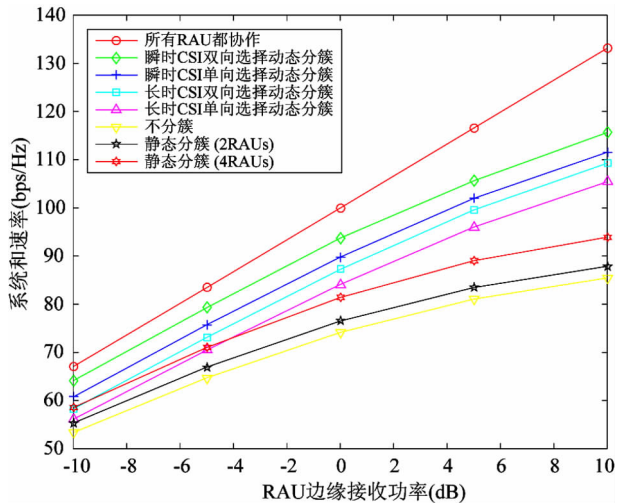


图 4 不同算法系统的和速率($K=10$)

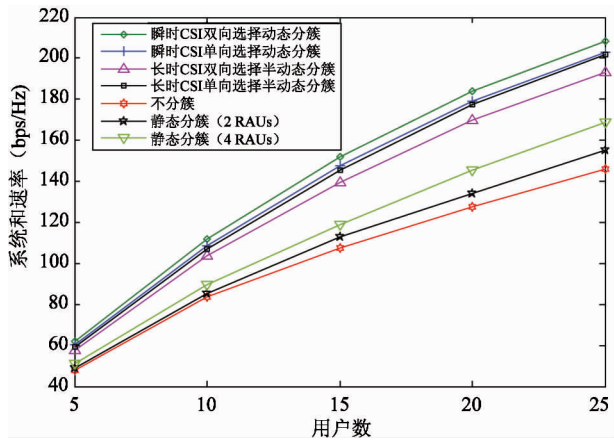


图5 不同用户数下的系统和速率

参考文献

- [1] Rusek F, Persson D, Lau B K, et al. Scaling up MIMO: opportunities and challenges with very large arrays [J]. *IEEE signal processing magazine*, 2013, 30(1): 40-60
- [2] Zheng M A, Zhang Z Q, Ding Z G, et al. Key techniques for 5G wireless communications: network architecture, physical layer, and MAC layer perspectives [J]. *Science China*, 2015, 58(4):41301-041301
- [3] 高翔, 孙强, 徐晨, 等. 静态分簇下的大规模分布式天线系统用户调度算法[J]. *电讯技术*, 2016, 56(10): 1075-1080
- [4] Yoon M, Kim M S, Lee C. A dynamic cell clustering algorithm for maximization of coordination Gain in uplink coordinated system[J]. *IEEE Transactions on Vehicular Technology*, 2016, 65(3):1752-1760
- [5] Papadogiannis A, Gesbert D, Hardouin E. A dynamic clustering approach in wireless networks with multi-cell cooperative processing[C]. In: *Proceedings of the IEEE International Conference on Communications*, Beijing, China, 2008. 4033-4037
- [6] Lu S H, Chen Y G, Wang L C. Antenna clustering for distributed large-scale MIMO systems[C]. In: *Proceedings of the IEEE International Conference on Internet of Things and Green Computing and Communications*, Taipei, China, 2014. 578-582
- [7] Lo T K Y. Maximum ratio transmission[C]. In: *Proceedings of the IEEE International Conference on Communications*, Vancouver, Canada, 1999. 1310-1314
- [8] Yoo T, Goldsmith A. Optimality of zero-forcing beamforming with multiuser diversity[C]. In: *Proceedings of the IEEE International Conference on Communications*, Seoul, Korea, 2005. 542-546
- [9] Ashikhmin A, Marzetta T L, Li L. Interference reduction in multi-cell massive MIMO systems I: large-scale fading precoding and decoding[EB/OL]. <http://arxiv.org/abs/1411.4182>; arxiv, 2014

Bidirectional semi-dynamic clustering for large-scale distributed antenna system

Tang Wei*, Sun Qiang***, Gao Xiang***, Wang Jue*, Li Yuhui*, Zhao Xinyu*

(* School of Electronic and Information, Nantong University, Nantong 226019)

(** Nantong Research Institute for Advanced Communication Technologies, Nantong 226019)

(*** Changzhou Power Supply Company, State Grid Jiangsu Electric Power Company, Changzhou 213000)

Abstract

As many existing algorithms depend on instantaneous channel state information (CSI) under the assumption of ideal condition, it is difficult to share all users' instantaneous CSI due to the limitation of backhaul capacity, a bi-directional semi-dynamic clustering algorithm is proposed. This algorithm uses long-term CSI, which reduces backhaul burden. Besides, it considers bidirectional clustering among remote antenna units, which could solve the limited-capacity problem caused by conventional unidirectional clustering scheme. Moreover, it uses greedy method to find the suboptimal cooperative cluster under the limitation of cluster size and thus reduce the computation complexity. Simulation results show that the proposed algorithm can achieve higher spectrum efficiency than the static clustering and is closed to the dynamic clustering algorithm using instantaneous CSI.

Key words: large-scale distributed antenna system, clustering algorithm, channel state information (CSI), spectrum efficiency