

## 基于协同度的基站群利益树动态分簇算法

黄开枝\* 郑丽清 李坤 吉江

(国家数字交换系统工程技术研究中心 郑州 450002)

**摘要:** 该文针对协同基站群分簇算法缺乏通用模型的问题,提出了一种协同度分簇模型,将系统和容量最大化简化为协同度最大化。在该模型的指导下,将分簇问题建模为有向带权连通图的利益树生成问题,设计了一种利益树动态分簇算法。该算法能够并行生成多个规模动态变化的协同簇,克服了传统顺序分簇导致的系统性能受限的问题;且分簇结果的协同度之和最大,可获得近似最优的分簇性能。仿真结果表明,该算法与传统贪婪搜索算法相比,系统频谱利用率提高了约 0.4 bit/Hz,且算法复杂度只与基站个数呈线性关系。

**关键词:** 无线通信;分簇;多基站协同;小区间干扰;信道状态信息

中图分类号: TN929.5

文献标识码: A

文章编号: 1009-5896(2012)06-1469-07

DOI: 10.3724/SP.J.1146.2011.00674

## Benefit-tree Dynamic Clustering Algorithm Based on Degree of Williness to Cooperate for Base Station Cooperation

Huang Kai-zhi Zheng Li-qing Li Kun Ji Jiang

(National Digital Switching System Engineering & Technological Research Center, Zhengzhou 450002, China)

**Abstract:** As the existing clustering algorithms are lack of effective guidable clustering model, an Degree of Williness to Cooperate (DWC) based clustering model is proposed, in which the clustering objective of maximizing the system sum rate is approximately to maximizing the sum of DWC between every two Base Stations (BS) in system. Based on this, the clustering issue is modeled as constructing benefit-trees of a connected graph with edge costs. Then a benefit-tree dynamic clustering algorithm is proposed. This algorithm simultaneously generates several clusters of dynamic size which could solve the limited-capacity problem caused by conventional orderly clustering scheme. Besides, the maximum sum of DWC in clustering results offers the approximately best system clustering capacity. Simulation results show that compared to the conventional greedy clustering algorithm, the system spectrum efficiency in this algorithm increases about 0.4 bit/Hz and the computational complexity is directly proportional to system size.

**Key words:** Wireless communication; Cluster; Base Stations (BS) cooperation; Inter-Cell Interference (ICI); Channel State Information (CSI)

### 1 引言

多基站协同处理技术<sup>[1]</sup>能够有效抑制小区间干扰(Inter-Cell Interference, ICI),提高系统频谱利用率,因而已经成为无线传输技术研究的热点<sup>[2,3]</sup>。但是,该技术需要基站间共享全部或部分的信道状态信息(Channel State Information, CSI)以及大量的用户数据信息<sup>[4]</sup>,所以当系统规模较大时,将给基站间的传输网络带来难以负荷的信息交互量<sup>[5,6]</sup>。为解决此问题,文献[7,8]采用分簇的方法,将整个系统划分为多个小的协同簇(coordination cluster),并令簇内的基站进行协同处理。现有的分簇算法主要分

为静态分簇算法及动态分簇算法。其中,文献[9,10]分别针对通信链路上、下行,将整个网络静态划分为多个协同簇,并利用协同处理技术消除簇内干扰(intra-cluster interference),提高了频谱利用率。但是这种静态分簇方式无法适应实际系统中信道的时变性,会损失一定的系统容量,因此近两年出现了动态分簇算法。文献[11-13]利用实时的CSI来设计动态分簇算法,获得了比静态分簇方案更好的系统性能。但是现有的动态分簇算法研究仍存在以下问题:

(1)缺乏可指导算法设计的分簇模型。现有文献中的动态分簇算法通常是根据某一设计原则对算法本身进行描述,均没有对分簇过程中如何选择协同基站与系统性能之间的关系进行讨论。

(2)顺序性及单向性基站选择破坏了最佳的分簇结构。现有分簇算法通过顺序生成的方式来产生多个协同簇,会导致后续生成的协同簇由于可候选

2011-07-03 收到, 2012-03-01 改回

国家科技重大项目(2008ZX03003-004)和国家973计划项目(2007CB310608)资助课题

\*通信作者: 黄开枝 huangkai-zhi@tsinghua.org.cn

的协作基站数减少而性能较差。

针对上述两个问题,本文首先引入了协同度的概念,定义协同度为基站间相互合作给双方带来的信干噪比增益,并以此来刻画基站间相互合作的愿望程度;然后,将系统和容量最大化近似等价于所有接收信干噪比(SINR)之和最大;通过分析基站端分簇之前及分簇之后的接收 SINR 之间的关系,得出所有接收 SINR 之和的最大化即为分簇后系统内协同度之和的最大化。基于此,搭建了协同度分簇模型。在该模型的指导下,将协同基站群的分簇问题建模为有向带权连通图的最大利益树生成问题,并设计了一种利益树动态分簇算法。该算法视每个基站为一棵树,并利用基站间双向协同度之和来定义两棵树间的利益度;然后,通过迭代过程不断选择利益度最大的两棵树进行合并,能够并行生成多个规模动态变化的协同簇。

在协同度模型指导下,利益树动态分簇算法克服了传统顺序分簇导致的系统性能受限的问题,能够获得近似最优的分簇性能。仿真结果表明:与传统的贪婪搜索动态分簇算法<sup>[11]</sup>相比,采用利益树分簇算法的系统频谱利用率提高了约 0.4 bit/Hz,且算法复杂度与基站个数  $N$  及簇规模限制参数  $K$  的对数成正比  $O(\lg(K) \cdot N)$ 。

## 2 系统模型

**定义 1** 协同基站群 设有  $N$  个基站,基站间采用有线高速链路(比如光纤)相连。两两基站间利用有线传输链路交互信息,并采用协同处理技术消除 ICI。将这  $N$  个基站视为一个协同基站群。

考虑一个有  $N$  个基站的协同基站群,基站服务的用户随机分布在小区边缘。对此系统进行分簇,系统模型如图1所示。用集合  $C = \{c_1, c_2, \dots, c_l\}$  表示分簇结果,  $|C| = l$  为协同簇的个数,符号  $|\cdot|$  表示集合内元素个数;  $c_i = [b_1, \dots, b_k, \dots, b_{|c_i|}]$ ,  $i = 1, \dots, l$  表示第  $i$  个协同簇内的基站集合,  $b_k$  为  $c_i$  中的第  $k$  个基站。将集合  $C$  映射到用户端,即令  $C \rightarrow U$ ,  $|U| = |C|$ , 集合  $U = \{u_1, u_2, \dots, u_l\}$  表示分簇结果对应的协同用户集合。其中  $c_i \rightarrow u_i$ ,  $|u_i| = |c_i|$ ,  $u_i = \{v_1, \dots, v_k, \dots, v_{|u_i|}\}$  表示第  $i$  个协同簇内的用户集合,  $v_k$  为基站  $b_k$  服务的用户,即  $b_k \rightarrow v_k$ 。

对于集合  $c_i$  内的所有基站,其接收信号可表示为

$$\mathbf{y}(c_i) = \mathbf{H}(u_i, c_i) \mathbf{A}(u_i) \mathbf{s}(u_i) + \sum_{u \neq u_i, u \in U} \mathbf{H}(u, c_i) \mathbf{A}(u) \mathbf{s}(u) + \mathbf{n}(c_i) \quad (1)$$

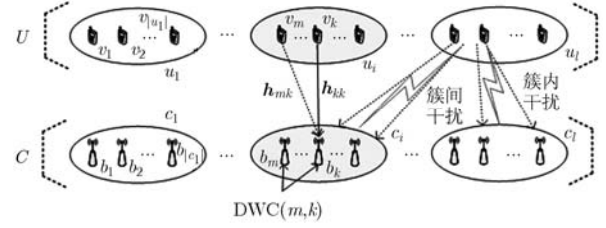


图1 系统模型

其中  $\mathbf{H}(u, c)$ ,  $u \in U, c \in C$  表示从用户集  $u$  到基站集  $c$  的信道参数矩阵;  $\mathbf{s}(u)$  表示用户集  $u$  的单位发射符号向量,有  $E[\mathbf{s}(u)\mathbf{s}^H(u)] = \mathbf{I}_{|s(u)|}$ , 其中  $\mathbf{I}_n$  表示  $n$  维单位矩阵;  $\mathbf{A}(u)$  表示用户端发射功率分配矩阵,设用户端等功率分配,则  $\mathbf{A}(u) = \sqrt{p} \times \mathbf{I}_{|u|}$ ,  $p$  为发射功率常量;  $\mathbf{n}(c_i)$  为均值为 0 的高斯白噪声向量,  $E[\mathbf{n}(c_i)\mathbf{n}^H(c_i)] = \sigma^2 \mathbf{I}_{|n(c_i)|}$ ,  $\sigma^2$  为噪声方差。

假设  $c_i$  内基站间不采用协同处理技术,对于第  $k$  个基站  $b_k$ ,其接收 SINR 为

$$\text{SINR}_k^{\text{non-coop}} = \frac{|h_{kk}|^2}{\sum_{v_j \neq v_k, v_j \in u_i} |h_{kj}|^2 + \sum_{u \neq u_i, u \in U} \sum_{v_n \in u} |h_{kn}|^2 + (\sigma^2)/p} \quad (2)$$

其中  $h_{kj}$  表示用户  $v_j$  到基站  $b_k$  的信道参数向量,  $v_n$  为  $u$  内第  $n$  个用户。

假设  $c_i$  内基站间采用迫零技术进行协同处理,即令  $\mathbf{W}(c_i)\mathbf{H}(u_i, c_i) = \mathbf{I}_{|c_i|}$ , 则

$$\mathbf{W}(c_i) = [\mathbf{H}^H(u_i, c_i)\mathbf{H}(u_i, c_i)]^{-1}\mathbf{H}^H(u_i, c_i) \quad (3)$$

其中  $\mathbf{W}(c_i) = [\mathbf{w}_1, \dots, \mathbf{w}_m, \dots, \mathbf{w}_{|c_i|}]^T$  为迫零处理矩阵。令  $\tilde{\mathbf{y}}(c_i)$  为经过处理后的信号,有

$$\tilde{\mathbf{y}}(c_i) = \mathbf{W}(c_i)\mathbf{y}(c_i) \quad (4)$$

将式(1)代入式(4),

$$\tilde{\mathbf{y}}(c_i) = \mathbf{W}(c_i)\mathbf{H}(u_i, c_i)\mathbf{A}(u_i)\mathbf{s}(u_i) + \mathbf{W}(c_i) \cdot \sum_{u \neq u_i, u \in U} \mathbf{H}(u, c_i)\mathbf{A}(u)\mathbf{s}(u) + \mathbf{W}(c_i)\mathbf{n}(c_i) \quad (5)$$

因此,基站  $b_k$  的接收 SINR<sup>[11]</sup>为

$$\text{SINR}_k^{\text{coop}} = \frac{|w_k h_{kk}|^2}{\sum_{v_j \neq v_k, v_j \in u_i} |w_k h_{kj}|^2 + \sum_{u \neq u_i, u \in U} \sum_{v_n \in u} |w_k h_{kn}|^2 + (|w_k|^2 \sigma^2)/p} \quad (6)$$

其中

$$\sum_{v_j \neq v_k, v_j \in u_i} |w_k h_{kj}|^2 \quad \text{及} \quad \sum_{u \neq u_i, u \in U} \sum_{v_n \in u} |w_k h_{kn}|^2$$

分别表示簇内干扰及簇间干扰。根据式(3),采用迫

零处理后，簇内干扰可被完全消除，即

$$\left. \begin{aligned} |\mathbf{w}_k \mathbf{h}_{kk}|^2 &= 1 \\ \sum_{v_j \neq v_k, v_j \in u_i} |\mathbf{w}_k \mathbf{h}_{kj}|^2 &= 0 \end{aligned} \right\} \quad (7)$$

因此，将式(7)代入式(6)，可得

$$\text{SINR}_k^{\text{coop}} = \frac{1}{\sum_{u \neq u_i, u \in U} \sum_{v_n \in u} |\mathbf{w}_k \mathbf{h}_{kn}|^2 + (|\mathbf{w}_k|^2 \sigma^2)/p} \quad (8)$$

### 3 基于协同度的分簇问题描述

#### 3.1 协同度定义

设有两个基站A和B。对于基站A，基站B与其合作给A带来的信干噪比增益越大，A与B合作的愿望应该越强。因此，定义基站 $b_k$ 与系统内其它任一基站(比如 $b_m$ )间的协同度 $\text{DWC}(k, m)$ 为基站 $b_k$ 与基站 $b_m$ 协作给基站 $b_k$ 带来的接收SINR增益，即

$$\text{DWC}(k, m) = \text{SINR}_k^{\text{coop}(k, m)} / \text{SINR}_k^{\text{non-coop}} \quad (9)$$

其中 $\text{SINR}_k^{\text{coop}(k, m)}$ 表示基站 $b_k$ 与基站 $b_m$ 协作时的接收SINR， $\text{SINR}_k^{\text{non-coop}}$ 表示基站 $b_k$ 没有与任何基站协作时的接收SINR。假设 $c_i$ 中只包含基站 $b_k$ 和基站 $b_m$ ，并且基站间采用迫零处理技术，则将式(2)和式(8)代入式(9)，可得 $\text{DWC}(k, m)$ 为

$$\begin{aligned} \text{DWC}(k, m) &= \frac{\left( |\mathbf{h}_{km}|^2 + \sum_{u \neq u_i, u \in U} \sum_{v_n \in u} |\mathbf{h}_{kn}|^2 + (\sigma^2)/p \right)}{|\mathbf{h}_{kk}|^2 \left( \sum_{u \neq u_i, u \in U} \sum_{v_n \in u} |\mathbf{w}_k \mathbf{h}_{kn}|^2 + (|\mathbf{w}_k|^2 \sigma^2)/p \right)} \\ &= \frac{\left( |\mathbf{h}_{km}|^2 + \sum_{u \neq u_i, u \in U} \sum_{v_n \in u} |\mathbf{h}_{kn}|^2 + (\sigma^2)/p \right)}{|\mathbf{w}_k \mathbf{h}_{kk}|^2 \left( \sum_{u \neq u_i, u \in U} \sum_{v_n \in u} |\mathbf{h}_{kn}|^2 + (\sigma^2)/p \right)} \\ &= 1 + \frac{|\mathbf{h}_{km}|^2}{\left( \sum_{u \neq u_i, u \in U} \sum_{v_n \in u} |\mathbf{h}_{kn}|^2 + (\sigma^2)/p \right)} \quad (10) \end{aligned}$$

#### 3.2 协同度分簇模型

在图1所示的系统模型中，系统的和容量为

$$R_{\text{coop}}(C) = \sum_{i=1}^l \sum_{v_k \in c_i} R_k^{\text{coop}} = \sum_{i=1}^l \sum_{v_k \in c_i} (\log_2(1 + \text{SINR}_k^{\text{coop}})) \quad (11)$$

可以看出，SINR的大小直接决定了系统和容量的大小。因此，可将和容量最大化简化为所有基站的接收SINR之和最大，即

$$\max \{R_{\text{coop}}(C)\} \rightarrow \max \left\{ \sum_{i=1}^l \sum_{v_k \in c_i} (\text{SINR}_k^{\text{coop}}) \right\} \quad (12)$$

令协同增益 $\Gamma_k^{\text{coop}}$ 表示整个系统分簇前与分簇后基站 $c_i^k$ 接收SINR的比值，即

$$\Gamma_k^{\text{coop}} = \text{SINR}_k^{\text{coop}} / \text{SINR}_k^{\text{non-coop}} \quad (13)$$

其中 $\text{SINR}_k^{\text{coop}}$ 表示系统分簇后，基站 $b_k$ 与簇 $c_i$ 内的其他所有基站协同处理时的接收SINR。

为了寻求接收SINR与协同度之间的关系，首先从协同增益入手，将式(2)和式(8)代入式(13)，

$$\begin{aligned} \Gamma_k^{\text{coop}} &= \frac{\sum_{v_j \neq v_k, v_j \in u_i} |\mathbf{h}_{kj}|^2 + \sum_{u \neq u_i, u \in U} \sum_{v_n \in u} |\mathbf{h}_{kn}|^2 + (\sigma^2)/p}{|\mathbf{h}_{kk}|^2 \left( \sum_{u \neq u_i, u \in U} \sum_{v_n \in u} |\mathbf{w}_k \mathbf{h}_{kn}|^2 + (|\mathbf{w}_k|^2 \sigma^2)/p \right)} \\ &= \frac{\sum_{v_j \neq v_k, v_j \in u_i} |\mathbf{h}_{kj}|^2 + \sum_{u \neq u_i, u \in U} \sum_{v_n \in u} |\mathbf{h}_{kn}|^2 + (\sigma^2)/p}{|\mathbf{h}_{kk}|^2 |\mathbf{w}_k|^2 \left( \sum_{u \neq u_i, u \in U} \sum_{v_n \in u} |\mathbf{h}_{kn}|^2 + (\sigma^2)/p \right)} \quad (14) \end{aligned}$$

将式(7)代入式(14)，可得

$$\begin{aligned} \Gamma_k^{\text{coop}} &= \frac{\sum_{v_j \neq v_k, v_j \in u_i} |\mathbf{h}_{kj}|^2 + \sum_{u \neq u_i, u \in U} \sum_{v_n \in u} |\mathbf{h}_{kn}|^2 + (\sigma^2)/p}{\left( \sum_{u \neq u_i, u \in U} \sum_{v_n \in u} |\mathbf{h}_{kn}|^2 + (\sigma^2)/p \right)} \\ &= 1 + \frac{\sum_{v_j \neq v_k, v_j \in u_i} |\mathbf{h}_{kj}|^2}{\sum_{u \neq u_i, u \in U} \sum_{v_n \in u} |\mathbf{h}_{kn}|^2 + (\sigma^2)/p} \\ &= 1 + \sum_{v_m \neq v_k, v_m \in u_i} \frac{|\mathbf{h}_{km}|^2}{\sum_{u \neq u_i, u \in U} \sum_{v_n \in u} |\mathbf{h}_{kn}|^2 + (\sigma^2)/p} \\ &= 2 - |u_i| \\ &\quad + \sum_{v_m \neq v_k, v_m \in u_i} \left( 1 + \frac{|\mathbf{h}_{km}|^2}{\sum_{u \neq u_i, u \in U} \sum_{v_n \in u} |\mathbf{h}_{kn}|^2 + (\sigma^2)/p} \right) \quad (15) \end{aligned}$$

由 $|u_i| = |c_i|$ 及式(10)，并将 $u_i \rightarrow c_i$ ， $v_m \rightarrow b_m$ ，可将式(15)替换为

$$\Gamma_k^{\text{coop}} = 2 - |c_i| + \sum_{v_m \neq v_k, v_m \in c_i} \text{DWC}(k, m) \quad (16)$$

其中 $\sum_{v_m \neq v_k, v_m \in c_i} \text{DWC}(k, m)$ 表示基站 $v_k$ 到 $c_i$ 内其他基站的协同度之和。

由式(13)知 $\text{SINR}_k^{\text{coop}} = \text{SINR}_k^{\text{non-coop}} \cdot \Gamma_k^{\text{coop}}$ ，将式(16)代入，所有信干噪比之和可等价

$$\begin{aligned}
 & \sum_{i=1}^l \sum_{v_k \in c_i} (\text{SINR}_k^{\text{coop}}) \\
 &= \sum_{i=1}^l \left( \sum_{v_k \in c_i} \text{SINR}_k^{\text{non-coop}} \cdot \Gamma_k^{\text{coop}} \right) \\
 &= \sum_{i=1}^l \left( \sum_{v_k \in c_i} \text{SINR}_k^{\text{non-coop}} \right. \\
 &\quad \cdot \left. \left( 2 - |c_i| + \sum_{v_m \neq v_k, v_m \in c_i} \text{DWC}(k, m) \right) \right) \\
 &= \sum_{i=1}^l \left( (2 - |c_i|) \cdot \sum_{v_k \in c_i} \text{SINR}_k^{\text{non-coop}} \right. \\
 &\quad \left. + \sum_{v_k \in c_i} \sum_{v_m \neq v_k, v_m \in c_i} \text{SINR}_k^{\text{non-coop}} \cdot \text{DWC}(k, m) \right) \quad (17)
 \end{aligned}$$

其中  $\text{SINR}_k^{\text{non-coop}}$  为未采用协同处理时基站  $k$  的接收 SINR。由于用户都处于小区边缘，故基站端接收 SINR 的值相似且比较低。因此设  $\text{SINR}_1^{\text{non-coop}} = \text{SINR}_2^{\text{non-coop}} = \dots = \text{SINR}_N^{\text{non-coop}}$ ，则式(17)继续简化为

$$\begin{aligned}
 & \sum_{i=1}^l \sum_{v_k \in c_i} (\text{SINR}_k^{\text{coop}}) \\
 &= (2 - |c_i|) \sum_{i=1}^l \left( \sum_{v_k \in c_i} \text{SINR}_k^{\text{non-coop}} \right) \\
 &\quad + \text{SINR}_k^{\text{non-coop}} \cdot \sum_{i=1}^l \left( \sum_{v_k \in c_i} \sum_{v_m \neq v_k, v_m \in c_i} \text{DWC}(k, m) \right) \quad (18) \\
 &\text{令 } \alpha = (2 - |c_i|) \sum_{i=1}^l \left( \sum_{v_k \in c_i} \text{SINR}_k^{\text{non-coop}} \right), \quad \beta = \\
 &\text{SINR}_k^{\text{non-coop}}, \quad \alpha, \beta \text{ 为常数, 则} \\
 &\quad \sum_{i=1}^l \sum_{v_k \in c_i} (\text{SINR}_k^{\text{coop}}) \\
 &= \alpha + \beta \cdot \sum_{i=1}^l \left( \sum_{v_k \in c_i} \sum_{v_m \neq v_k, v_m \in c_i} \text{DWC}(k, m) \right) \quad (19)
 \end{aligned}$$

因此，接收 SINR 最大化等价于系统内基站间协同度之和的最大化

$$\begin{aligned}
 & \max \{R_{\text{coop}}(C)\} \rightarrow \max \left( \sum_{k=1}^N \text{SINR}_k^{\text{coop}} \right) \\
 &\rightarrow \max \left( \sum_{i=1}^l \left( \sum_{v_k \in c_i} \sum_{v_m \neq v_k, v_m \in c_i} \text{DWC}(k, m) \right) \right) \quad (20)
 \end{aligned}$$

### 3.3 基于协同度的分簇问题描述

利用有向带权连通图  $G = [V, \mathbf{W}]$  来描述一个有  $N$  个基站的协同基站群。其中， $V$  为图  $G$  的顶点集合，令每个基站为一个顶点，有  $|V| = N$ ； $\mathbf{W} \in \mathbb{C}^{N \times N}$  表示图  $G$  的权值矩阵，利用协同度来描述权值矩阵  $\mathbf{W}$ 。令  $w(k, j) \in \mathbf{W}$  为节点  $k$  和  $j$  间的权值，则

$$w(k, j) = \begin{cases} \text{DWC}(k, j), & j \neq k \\ 0, & j = k \end{cases} \quad (21)$$

用一个森林  $F$  来描述协同基站群的分簇结果  $C$ 。令  $F = \{t_1, t_2, \dots, t_i, \dots, t_{N_T}\}$ ，其中， $N_T$  为森林中树的棵数， $t_i$  为森林中的第  $i$  棵树。则可以知道， $C \rightarrow F, c_i \rightarrow t_i$  且  $N_T = l$ 。

**定义 2** 树的利益 假设森林  $F$  中的每棵树是带有利益的利益树，并令  $\Gamma(t_i)$  为树的利益，将其定义为  $t_i$  内节点间的权值之和，即

$$\Gamma(t_i) = \sum_{k \in t_i} \sum_{j \in t_i, j \neq k} w(k, j) \quad (22)$$

**定义 3** 森林的利益 将包含利益树的森林称为利益森林。其中，森林的利益  $\Gamma(F)$  定义为森林中每棵树的利益之和。即

$$\Gamma(F) = \sum_{i=1}^{N_T} \Gamma(t_i) \quad (23)$$

将式(21)代入式(22)并代入式(23)，则

$$\Gamma(F) = \sum_{i=1}^{N_T} \left( \sum_{k \in t_i} \sum_{j \in t_i, j \neq k} \text{DWC}(k, j) \right) \quad (24)$$

对比式(24)和式(20)可以看出，森林的利益与协同度之和是一致的。因此，可以将分簇问题转化为连通图  $G$  的最大利益树生成问题。即已知一个有向带权连通图  $G$ ，如何生成一个森林  $F^*$ ，使得森林中每棵树中节点个数不能超过  $K$ ，且森林的利益最大。具体描述如下：

$$\left. \begin{aligned} & \Gamma(F^*) = \max_F \{ \Gamma(F) \} \\ & \max_{t_i \in F^*} (|t_i|) \leq K \end{aligned} \right\} \quad (25)$$

其中  $K$  为协同簇规模限制参数，表示分簇结果中协同簇中的基站个数不得超过  $K$ 。

### 4 最大利益树分簇算法

**定义 4** 利益度 用利益度  $C(t_m, t_n)$ ， $m \neq n$  来描述树  $t_m$  与树  $t_n$  进行合并带来的协同度量；定义  $C(t_m, t_n)$  为树  $t_m$  中的节点与树  $t_n$  中的节点相互间的权值之和的归一化。即

$$C(t_m, t_n) = \frac{1}{(2 \cdot |t_m| \cdot |t_n|)} \left\{ \sum_{k \in t_m} \sum_{j \in t_n} (w(k, j) + w(j, k)) \right\} \quad (26)$$

由于树的节点数越多，权值之和越大。因此为了消除树的规模对利益度的影响，在式(26)中进行了归一化。从式(26)中可看出，利益度消除了协同度的方向性，这样可以避免当两个基站间的协同度相差很大时，即一方合作的愿望很强，另一方合作的愿望很弱时，单向选择导致系统性能较差的问题。

基于上述分析，设计了利益树分簇算法：首先，

利用树内节点间双向的权值之和来定义树与树之间的利益度；然后，通过迭代过程不断选择利益度最大的两棵树进行合并，并更新森林中其他树与新生成树间的利益度；直到森林中每两棵树间的利益度之和为零算法结束。

(1)算法初始化阶段 将图  $G$  转化为一个初始森林  $F^0$ ，其中， $F^0$  中每棵树只含一个节点，每个节点为独立的一棵树， $N_T = N$ 。根据式(26)，初始森林中任两棵树  $t_m$  和  $t_n$  间的协同度可计算为

$$C(t_m, t_n) = (w(m, n) + w(n, m))/2 \quad (27)$$

(2)算法迭代阶段

步骤 1 选出相互协同度最大的两棵树  $t'_m, t'_n$ ，将它们合并为一棵新树  $t_{\text{new}}$ 。每合并一次，森林中的树减少一颗。

$$C(t'_m, t'_n) = \max_F \{C(t_m, t_n)\} \quad (28)$$

$$t_{\text{new}} = t'_m \cup t'_n \quad (29)$$

$$N_T \leftarrow N_T - 1 \quad (30)$$

步骤 2 按照式(31)更新其它树与新树  $t_{\text{new}}$  之间的利益度。如果两棵树中的节点个数之和超过  $K$ ，则令两棵树间的利益度为零。

$$C(t_m, t_n) = \begin{cases} \sum_{k \in t_m} \sum_{j \in t_n} (w(k, j) + w(j, k)) / (2 \cdot |t_m| \cdot |t_n|), & |t_m| + |t_n| \leq K \\ 0, & \text{其它} \end{cases} \quad (31)$$

步骤 3 判断森林中每两颗树间的利益度之和是否为零。为零则算法结束，输出分簇结果，否则返回步骤 1。

在算法迭代中，由于树与树之间不断合并，树的规模越来越大。因此，森林中会逐渐出现两棵树的节点个数之和大于  $K$  的情况，此时根据式(31)将其对应的利益度置为零。最终所有树间的利益度都为零，算法结束。可以看出，本算法在生成协同簇时是并行生成的，并没有按顺序逐一生成协同簇，这种方式能够克服现有算法顺序分簇导致的协同簇性能不平衡问题，提高了系统整体性能。

定理 1 说明了利益树分簇算法的有效性。

**定理 1** 利益树分簇算法能够在树规模限制条件下生成利益最大的森林  $F^*$ 。

**证明** 记  $F^i$  为第  $i$  次迭代时生成的森林， $N_{T_i} = |F^i|$  为第  $i$  次迭代时森林中树的棵数，有  $N_{T_i} = N - i$ ， $t'_m, t'_n$  为森林  $F^i$  中利益度最大的两棵树，令  $N^* = |F^*|$  为算法结束时森林中树的棵数。

对一个初始森林  $F^0$ ，森林中有  $N$  棵树，每棵树中只含一个节点。根据式(21)，由于  $w(i, i) = 0$ ，初

始森林中，每颗树的利益  $\Gamma(t_m) = 0, m = 1, \dots, N_{T_0}$ ，故此时森林的利益  $\Gamma(F^0) = 0$ 。

根据算法步骤 1，第 1 次迭代后， $F^0$  里两棵利益度最大的树  $t'_m, t'_n$  合并为一棵树  $t'_{\text{new}}$ ，此时

$$\Gamma(F^1) = \sum_{t \in F^0, t \neq t'_m, t \neq t'_n} \Gamma(t) + \Gamma(t'_{\text{new}}) \quad (32)$$

根据式(22)有新生成的树的利益  $\Gamma(t'_{\text{new}})$  为

$$\begin{aligned} \Gamma(t'_{\text{new}}) &= \sum_{k \in t'_m} \sum_{j \in t'_n, j \neq k} w(k, j) \\ &= \sum_{k \in t'_m} \sum_{j \in t'_n, j \neq k} w(k, j) \\ &\quad + \sum_{k \in t'_n} \sum_{j \in t'_m, j \neq k} w(k, j) \\ &\quad + \sum_{k \in t'_m} \sum_{j \in t'_n} (w(k, j) + w(j, k)) \end{aligned} \quad (33)$$

根据式(26)，式(33)可简化为

$$\Gamma(t'_{\text{new}}) = \Gamma(t'_m) + \Gamma(t'_n) + C(t'_m, t'_n) \cdot (2 \cdot |t'_m| \cdot |t'_n|) \quad (34)$$

将式(34)代入式(32)有

$$\begin{aligned} \Gamma(F^1) &= \sum_{t \in F^0, t \neq t'_m, t \neq t'_n} \Gamma(t) + \Gamma(t'_m) + \Gamma(t'_n) \\ &\quad + C(t'_m, t'_n) \cdot (2 \cdot |t'_m| \cdot |t'_n|) \\ &= \Gamma(F^0) + C(t'_m, t'_n) \cdot (2 \cdot |t'_m| \cdot |t'_n|) \\ &= C(t'_m, t'_n) \cdot (2 \cdot |t'_m| \cdot |t'_n|) \end{aligned} \quad (35)$$

因此，由式(26)及式(28)知  $F^1$  为森林棵数为  $(N - 1)$  时利益最大的森林。

若已知  $F^i$  为森林棵数为  $(N - i)$  时利益最大的森林，有  $F^i$  的利益为  $G(F^i)$ ，对第  $i + 1$  次迭代，

$$\begin{aligned} \Gamma(F^{i+1}) &= \sum_{t \in F^i, t \neq t'_m, t \neq t'_n} \Gamma(t) + \Gamma(t'_{\text{new}}) \\ &= \sum_{t \in F^i, t \neq t'_m, t \neq t'_n} \Gamma(t) + \Gamma(t'_m) + \Gamma(t'_n) \\ &\quad + C(t'_m, t'_n) \cdot (|t'_m| \cdot |t'_n|) \\ &= \Gamma(F^i) + C(t'_m, t'_n) \cdot (|t'_m| \cdot |t'_n|) \end{aligned} \quad (36)$$

由于  $t'_m, t'_n$  是  $F^i$  中相互利益度最大的两颗树，即  $C(t'_m, t'_n) \cdot (|t'_m| \cdot |t'_n|)$  在  $F^i$  中的利益度最大，因此得出  $F^{i+1}$  为森林棵数为  $(N - i - 1)$  时利益最大的森林。

依此类推，可得  $F^*$  为森林棵数为  $N^*$  时利益最大的森林。其中，最大利益森林中的棵数  $N^*$  受参数  $K$  影响，故利益树分簇算法可以生成树规模限制条件下的最大利益森林。证毕

## 5 仿真结果及分析

本节利用蒙特卡洛方法对所提算法进行性能仿真。首先分析了利益树分簇算法的分簇结果；然后比较了利益树分簇算法与现有分簇算法的性能；最

后分析了算法的复杂度。

考虑上行通信链路，设协同基站群小区个数  $N = 61$ ，令小区半径  $R = 1 \text{ km}$ ，定义小区边缘区域半径为  $0.8R$  到  $R$  之间的环形区域。设每个小区边缘区域内随机分布一个用户，并令协同簇规模最大为  $K = 6$ 。

(1)系统在分簇时只考虑大尺度衰落参数的影响。即令信道参数  $h_{ij} = a_{ij}$ ， $a_{ij}$  表示大尺度衰落，有  $a_{ij}^2 = PL_{ij} \cdot \phi_{ij}$ 。其中  $\phi_{ij}$  为阴影衰落，服从对数正态分布，方差设为 8 dB， $PL_{ij}$  为路径衰落， $PL_{ij} = PL_0 \cdot d_{ij}^{-\lambda}$ ；其中， $d_{ij}$  表示用户  $i$  到基站  $j$  的距离， $\lambda$  为路径损耗因子设为 3.76<sup>[11]</sup>， $PL_0$  为路径衰落常数。

(2)系统分簇后，对分簇结果进行性能仿真时采用完整的信道状态信息。即  $h_{ij} = a_{ij} \cdot b_{ij}$ ，其中  $b_{ij}$  表示小尺度衰落信道参数，服从瑞利分布。

### 5.1 算法分簇结果分析

利用利益树分簇算法进行分簇，其分簇结果如图 2 所示。其中，图 2(a)为系统的分簇结果，星点表示基站，圆点表示小区内用户。图中相连的节点组成一棵利益树，即分簇结果中的一个协同簇。从图 2(a)中看出，分簇结果中协同簇的规模并不固定，而是受实际信道干扰情况而自适应调整，且不超过 6。图 2(b)给出了图 2(a)分簇结果中一棵树(协同簇)的生成过程。从图 2(b)可以看出，森林中利益树通过不停合并的过程达到收敛，而且分簇过程中，各个簇之间是并行生成，没有先后顺序，因此不存在顺序分簇存在的簇性能不均衡问题。

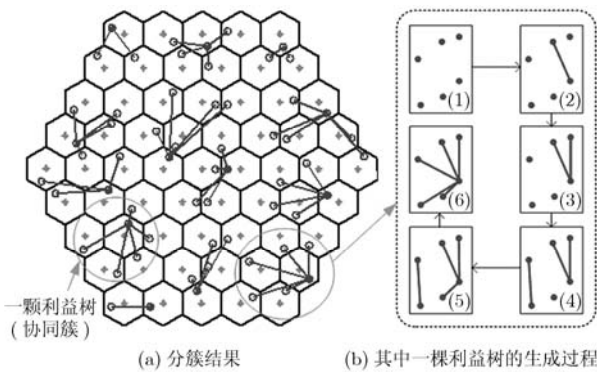


图 2 利益树动态分簇结果

### 5.2 算法性能比较

每一次仿真时重新生成一组用户的分布位置，然后采用利益树动态分簇算法进行分簇，并根据式 (11)计算系统的总容量。仿真 100 次后，对系统总容量取平均值，并平均到每个小区，即为每个小区的平均频谱利用率。图 3 给出了利益树分簇算法的性能仿真结果。其中，横坐标系统信噪比是忽略小区之间干扰后各基站的接收信噪比均值，纵坐标为

平均频谱利用率。为了便于比较，图 3 中还给出了无协同、静态、半动态<sup>[11]</sup>(在文献[12]中将文献[11]中的贪婪搜索算法定义为半动态分簇算法)以及完全动态<sup>[12]</sup>几种分簇算法的性能仿真结果。其中，无协同是指单基站无协作的服务模式；静态分簇算法将相邻 6 个基站划分为一个协同簇；半动态分簇是指簇规模大小固定为 6；完全动态方案是采用协同门限方式来判断基站间是否协同，即计算基站  $B$  服务的用户对基站  $A$  的干扰强度与基站  $A$  服务用户的信号强度的比值。如果比值大于设定的门限 0.1(仿真得出的最佳值)，则  $A$  和  $B$  协同。从图 3 中可知，所有动态分簇算法性能都优于静态分簇性能。且利益树动态分簇算法性能优于其它两种现有典型的动态分簇算法。相比于贪婪搜索算法，采用利益树分簇算法系统的频谱利用率提高了约 0.4 bit/Hz。

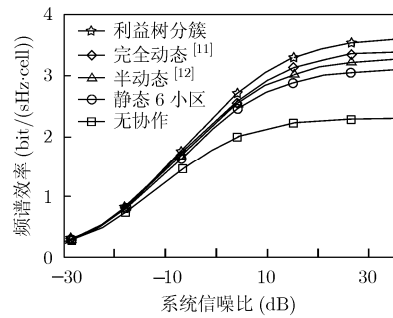


图 3 利益树分簇算法性能对比图

### 5.3 算法复杂度分析

从利益树分簇算法流程可知，每次迭代时森林中都有两棵树进行合并，因此本算法的迭代次数小于初始森林中树的棵数，即小于  $N$ 。图 4 给出了算法迭代次数与分簇规模限制参数  $K$  以及基站个数  $N$  的关系。其中图 4(a)为  $N = 200$  时迭代次数与  $K$  的关系曲线，利用曲线拟合方法可得迭代次数  $y = 176 \lg(K) + 3.7$ ；图 4(b)为  $K = 6$  时迭代次数与  $N$  的关系，利用曲线拟合可得迭代次数与  $N$  为  $y = 0.82N + 0.5$ 。因此，算法的时间复杂度与  $K$  的对数及基站个数  $N$  成线性关系，即为  $O(\lg(K) \cdot N)$ 。

贪婪算法是先从  $N$  个基站挑出一个基站，然后从剩余的  $N - 1$  个基站中选出一个使系统协同性能最好的基站，直到形成  $K$  大小的簇；之后，按以上方式从  $N - K$  个基站中再次形成一个簇，以此类推，用以上方式形成多个簇。因此，贪婪算法的时间复杂度为  $O(N!/K!)$ 。最优算法是穷举所有的分簇情况，选取系统协同性能最好的一种分簇情况作为分簇结果。最优算法的时间复杂度极高，为  $O(N! \cdot N!)$ 。表 1 列出了这 3 种分簇算法的时间复杂度比较，可以看出本文提出的利益树分簇算法的复杂度最低。

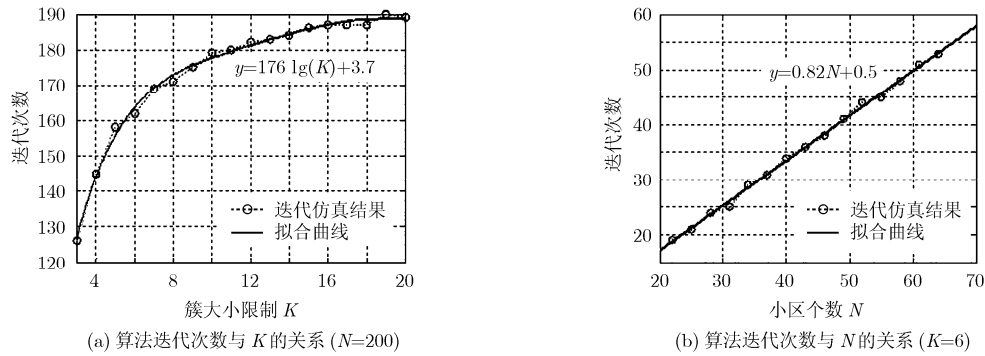


图 4 利益树分簇算法复杂度分析

表 1 分簇算法复杂度比较

算法类型	利益树算法	贪婪算法	最优算法
复杂度	$O(\lg(K) \cdot N)$	$O(N!/K!)$	$O(N! \cdot N!)$

## 6 结论

本文提出了一种协同度分簇模型，并在此模型指导下设计了一种并行生成协同簇的利益树动态分簇算法。主要的研究成果及结论如下：引入了协同度的概念，并将系统和容量最大化目标简化为协同度最大，搭建了协同度分簇模型；将分簇问题等价描述为：已知一个有向带权连通图，如何生成一个森林使得森林中每棵树中节点个数不得超过  $K$ ，且森林中所有树的利益之和最大。利益树分簇算法是并行生成多个协同簇的动态分簇算法，能够克服现有顺序分簇算法导致的协同簇性能不均衡问题，提高系统整体性能。理论推导证明了利益树算法能够产生利益最大的森林，而利益最大即为协同度之和最大，因此该算法是以协同度最大为目标的最佳分簇算法，可获得近似最优的分簇性能。仿真分析表明，该算法优于现有的动态分簇算法，相比于贪婪搜索算法，在本文仿真环境下，系统的频谱利用率提高了约 0.4 bit/Hz，且算法复杂度与  $K$  的对数及基站个数  $N$  成正比，为  $O(\lg(K) \cdot N)$ 。

## 参考文献

- [1] Shamai S and Zaidel B M. Enhancing the cellular downlink capacity via co-processing at the transmitting end [C]. IEEE Vehicular Technology Conference (VTC), Rhodes, Greece, 2001, 3: 1745-1749.
- [2] Gesbert D, Hanly S, Huang H, et al. Multi-cell MIMO cooperative networks: a new look at interference [J]. *IEEE Journal on Selected Areas in Communications*, 2010, 28(9): 1380-1408.
- [3] Nam Youngchan, Liu Lingjia, Wang Yan, et al. Cooperative communication technologies for LTE-advanced [C]. Acoustics Speech and Signal Processing (ICASSP), Dallas, TX, 2010: 5610-5613.
- [4] Seokhyun Y, Somekh O, Simeone O, et al. A comparison of opportunistic transmission schemes with reduced channel information feedback in OFDMA downlink [C]. Personal, Indoor and Mobile Radio Communications (PIMRC), Athens, 2007: 1-5.
- [5] Niu Bo, Simeone O, Somekh O, et al. Ergodic and outage performance of fading broadcast channels with 1-bit feedback [J]. *IEEE Transactions on Vehicular Technology*, 2010, 59(3): 1282-1293.
- [6] Papadogiannis A, Bang H J, Gesbert D, et al. Downlink overhead reduction for multi-cell cooperative processing enabled wireless networks [C]. Personal, Indoor and Mobile Radio Communications (PIMRC), Cannes, France, 2008: 1-5.
- [7] Hoydis J, Kobayashi M, and Debbah M. On the optimal number of cooperative base stations in network MIMO systems [J]. <http://arxiv.org/abs/1003.0332>, 2010.
- [8] Ng C and Huang H. Linear precoding in cooperative MIMO cellular Networks with limited coordination clusters [J]. *IEEE Journal on Selected Areas in Communications*, 2010, 28(9): 1446-1454.
- [9] Venkatesan S. Coordinating base stations for greater uplink spectral efficiency in a cellular network [C]. Personal, Indoor and Mobile Radio Communications (PIMRC), Athens, Greece, 2007: 13-18.
- [10] Boccardi F and Huang H. Limited downlink network coordination in cellular networks [C]. Personal, Indoor and Mobile Radio Communications (IEEE PIMRC), Athens, Greece, 2007: 1-5.
- [11] Papadogiannis A, Gesbert D, and Hardouin E. A dynamic clustering approach in wireless networks with multi-cell cooperative processing [C]. IEEE International Conference on Communications (ICC), Beijing, 2008: 4033-4037.
- [12] Liu Jing-xin and Wang Dong-ming. An improved dynamic clustering algorithm for multi-user distributed S antenna system [C]. Wireless Communications & Signal Processing (WCSP), Wuhan China, Nov. 2009: 1-5.
- [13] Zhou Sheng, Gong Jie, Niu Zhi-sheng, et al. A decentralized framework for dynamic downlink base station cooperation [C]. Global Telecommunications Conference (IEEE GLOBECOM), Nanjing China, 2009: 1-6.

黄开枝：女，1973年生，硕士生导师，研究方向为移动通信技术。  
 郑丽清：女，1987年生，硕士，研究方向为移动通信技术。  
 李坤：男，1987年生，硕士，研究方向为移动通信技术。  
 吉江：男，1983年生，博士，研究方向为移动通信技术。