On the Complexity of Minimum Partition of Frequency-Agile Radio Networks

V.S. Anil Kumar
Madhav V. Marathe
Department of Computer Science and
Virginia Bioinformatics Institute,
Virginia Tech, Blacksburg, VA 24061
Email: {akumar, mmarathe}@vbi.vt.edu

Sriram V. Pemmaraju
Department of Computer Science,
University of Iowa,
Iowa City, IA 52242
Email: sriram@cs.uiowa.edu

Imran A. Pirwani
Department of Computer Science,
University of Iowa,
Iowa City, IA 52242
Email: pirwani@cs.uiowa.edu

***Abstract*—New advances in cognitive radio technology, and the recent proposals for opening up the licensed spectrum bands raise new challenges for frequency allocation problems. In this paper, we study the *Clustering in Frequency-Agile Radios* problem (CFRP), which involves partitioning the network into the smallest number of connected clusters, where nodes within a cluster can communicate with a common frequency without creating interference to any primary user. This problem was formulated by Steenstrup (DySPAN '05) to model one-to-many broadcasts and Steenstrup's paper explored the empirical performance of greedy heuristics. In this paper, we focus on the formal computational complexity of this problem, and prove that it is NP-complete. Moreover, we show that it is unlikely that there exists a polynomial-time algorithm for this problem that always produces a solution (for n-vertex graphs) with at most $\ln n$ times the optimal number of components. We show that several natural greedy heuristics, including the one studied by Steenstrup, can have highly sub-optimal performance in the worst case. For trees, we show that the optimum solution can be computed in polynomial time.**

There are instances where the number of components required is very large, if we require them to be strictly disjoint. We observe that in practice, allowing even a small amount of overlap among clusters significantly reduces the number of clusters needed. Motivated by this, we study a relaxation of the CFRP problem, that allows components to overlap, and study bicriteria approximations for this version of the problem, by simultaneously bounding the number of components and the average overlap between them. We present efficient algorithms that produce solutions with the cost related to both of these metrics being within a factor of $O(\log n)$ of the optimum. We end with simulations results for our algorithms on a wide range of instances.

I. Introduction

Several studies have shown that the licensed spectrum bands are extremely underutilized, e.g., the utilization in some bands can vary between 15% to 85% [8]. As a result, there are several proposals for Cognitive Networks (also referred to as Dynamic Spectrum Access Networks [1]) to open up some bands to unlicensed users (henceforth referred to as *secondary users*), as long as they do not disrupt the operation of the licensed users (henceforth, referred to as *primary users*); interest in these proposals has grown because of significant advances in cognitive radio technology which is likely to lead to devices with improved capabilities to sense the spectrum usage, and switch channels and power levels with low overheads [1]. Since a key requirement for these networks is to make sure that primary users are not disrupted, they involve first sensing spectrum usage by primary users, and then allocating unused bands to secondary users. Our focus in this paper is on the latter step - we assume that the spectrum sensing steps are already done, and the primary users and the channels used by them are already known; as pointed out in [1], the problem of effectively allocating the channels to secondary users remains challenging even with this assumption.

Frequency allocation problems have been studied extensively as coloring problems in wireless networks [21]. We extend a formulation by Steenstrup [23] for opportunistic channel assignment in such networks. Formally, let V denote the set of secondary nodes. We assume a fixed transmission range for each node, and this results in a graph $G = (V, E)$, with $e = (u, v) \in E$ if u and v are within transmission range of each other. We assume that each node can switch among a set $\mathcal{F}$ of channels, and let F_u^T and F_u^R denote the sets of frequencies on which it is safe for node u to transmit and receive, respectively, without interference to the primary users; as mentioned earlier, we assume that these sets are already known as a result of the spectrum sensing step, using methods described in [1], [11], [25]. Because of non-uniform primary user constraints, there may be no common frequency on which all secondary users could communicate and reach each other, and some nodes might have to switch channels, in order to enable all nodes to be connected, which can have additional overheads. We formalize this by the *Clustering in Frequency-Agile Radios* problem (CFRP) (referred to as the *Broadcast Frequency Assignment* in [23], and defined formally in Section II), where the goal is to partition the nodes into the smallest number of connected clusters $S_1, \ldots, S_K$ so that nodes within each cluster can communicate using a common frequency, and a node u in cluster S_i can communicate with a neighbor $v \in S_j$ by switching to another permitted frequency. Reducing the number of clusters reduces the amount of channel switching that needs to be done, in order to let any pair of nodes to communicate. Steenstrup develops centralized and distributed algorithms for this problem, which are based on greedy heuristics, and studies their performance empirically, without any provable performance guarantees. The focus of this paper is to study the complexity of this problem formally, and our main results are the following.

1) We show that the greedy heuristic of [23] does not always give the optimum solution, as do other classes of greedy heuristics. We prove that the CFRP problem is NP-complete, and so it is unlikely that simple polynomial time heuristics would give the optimum solution. In fact, we show that this problem cannot be approximated efficiently within a factor $(1-o(1))\ln n$ unless NP $\subset$ DTIME$(n^{O(\log\log n)})$. In other words, there is unlikely to exist a polynomial time algorithm that would give a solution with $(1-o(1))\ln n\text{OPT}(\mathcal{I})$ clusters, for any arbitrary instance $\mathcal{I}$ of the CFRP problem, where OPT$(\mathcal{I})$ denotes the cost of the optimum solution for this instance.
2) We study a relaxed version of the CFRP problem, denoted by R-CFRP, in which the clusters are allowed to overlap. A node appearing in multiple clusters incurs a cost of switching between the common frequencies for these clusters, and we control this switching cost by minimizing the total overlap between clusters, in addition to the number of clusters. We develop a randomized algorithm that gives a solution with cost $O(\log n)$ relative to both these objectives, based on linear programming. Since linear programs are expensive to solve, we study combinatorial algorithms for this problem. We show that a natural greedy heuristic for this problem is not very efficient, but a variant of it based on the technique of Lagrangian multipliers gives logarithmic bounds on the performance.
3) Finally, we study the empirical performance of the greedy algorithms for the CFRP and R-CFRP problems, and show that allowing overlaps leads to significantly smaller number of clusters. In particular, we study the relative performance of two algorithms: one that repeatedly removes a largest connected component which is CFRP feasible (GREEDY), and our greedy variant for R-CFRP that allows some overlap between clusters (GREEDY2). We ran simulations on randomly generated feasible instances of the problem on randomly generated *unit disk graphs* (UDGs). Our simulation results show that, while allowing a small amount of overlap, GREEDY2 yields far fewer clusters compared to GREEDY. Specifically, our results range from the simple greedy algorithm yielding about 30 times more clusters than GREEDY2 for "large" graphs (with about 60,000 nodes uniformly and randomly distributed over a square region) to the simple greedy algorithm yielding about 1.5 times more clusters for "small" graphs (with about 6000 nodes) that are heavily clustered. The average overlap per vertex ranged from about 5 (for the "small" graphs) to about 10 (for the "large" graphs).

The focus of our results is a theoretical look at the complexity of various broadcast frequency assignment problems for cognitive networks. Since greedy heuristics are commonly used for such problems, our results suggest that a closer look at their worst case performance might give better insights into their performance.

II. Preliminaries and Notation

We follow the notation of [23] to the extent possible. Let V denote the set of secondary users, henceforth referred to as nodes. Let the transmission power levels of all the nodes be fixed, and let E denote the resulting set of edges, with $(v,w)\in E$ if w is within the range of v; we assume that the transmission ranges are such that the edges are all bidirectional. Let $G=(V,E)$ denote the resulting graph; we assume that this does not depend on the frequency of transmission. Let $N(v)=\{w:(v,w)\in E\}$ denote the set of neighbors of v. Let $\mathcal{F}$ denote the set of all frequencies that the nodes can use. For node $v\in V$, let P_v denote the set of frequencies being used by primary users within the transmission range of node v. Then, $F_v^R=\mathcal{F}-P_v$ denotes the set of frequencies on which node v can receive. $F_v^T=F_v^R\cap_{w\in N(v)}F_w^R$ denotes the set of nodes on which node v can transmit, and not cause interference to any receiver w in $N(v)$.

The *Clustering in Frequency-agile Radios* Problem (CFRP). Given an instance $\mathcal{I}$ of the CFRP problem, specified by the graph $G=(V,E)$, and the sets F_v^R for all $v\in V$, the objective is to partition V into K sets $\{V_1,\ldots,V_K\}$, and choose frequency f_i for $i=1,\ldots,K$, so that (i) K is minimized, (ii) $G[V_i]$ is connected for each i, and (iii) for each $i=1,\ldots,K$, and for each $v\in V_i$, $f_i\in F_v^T$. See Figure 1 for an illustration of this problem.

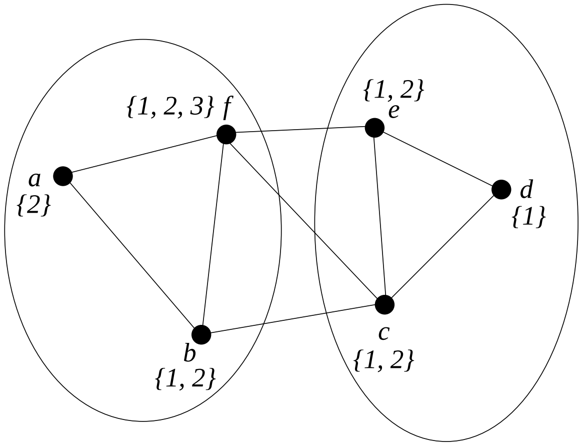

Fig. 1. This shows a graph in which each vertex is assigned a set of frequencies F^R. For each vertex $u\in\{a,b,f\}$, $F_u^T=\{2\}$ and for each vertex $u\in\{c,d,e\}$, $F_u^T=\{1\}$. Thus an optimal partition solution to CFRP is the partition $V_1=\{a,b,f\}$ and $V_2=\{c,d,e\}$.

A couple of remarks about this problem are in order. The common frequency f_i is the frequency at which communication with each connected component $G[V_i]$ takes place. If a node $u\in V_i$ needs to communicate with a neighboring node $v\in V_j$, $i\neq j$, then either u needs to switch frequencies from

f_i to f_j or v should switch from listening on f_j to listening on f_i.

In the *Relaxed* CFRP problem (R-CFRP), we allow the sets V_i to overlap, and the objective is to minimize $\sum_i |V_i|$, while ensuring that the number of sets chosen is at most an input parameter K. Note that $(\sum_i |V_i|)/|V|$ is the average number of sets V_i that a vertex appears in. In the CFRP problem, this ratio is 1, whereas in R-CFRP this can be larger than 1.

We will be interested in approximation algorithms with provable approximation guarantees for these problems, which is defined in the following manner. For the CFRP problem, we say that an algorithm is an α-approximation, if for any input instance $\mathcal{I}$, the algorithm provably chooses at most $\alpha\text{OPT}(\mathcal{I})$ sets, where $\text{OPT}(\mathcal{I})$ denotes the number of sets in the optimal solution. For the R-CFRP problem, we consider bicriteria approximations - an algorithm is said to produce an (α, β)-approximation, if it chooses at most $L = \alpha K$ sets $\{V_1, \ldots, V_L\}$ so that $\sum_i |V_i| \leq \beta\text{OPT}$, where OPT now denotes the cost of an optimum solution that chooses at most K sets.

A. Related Work

There has been a lot of research on various aspects of Dynamic Spectrum Access in recent years; see [2] for a good survey. There have been numerous proposals on orthogonal frequency allocation to avoid interference, and optimize system utility [4], [5], [17], [18], [20], [26]. Most of these assume static spectrum demand, and do not consider stability issues. This work is extended to fairly general dynamic demands in [6]. Steenstrup [23], [22] studies more general problems of broadcast frequency assignment, which is the problem we study here.

There has also been a lot of work on market driven approaches to spectrum sharing, in which secondary users have demand profiles for spectrum as a function of the clearing price, and a central agent has to allocate the spectrum so that it does not cause interference to the primary and secondary users. Gandhi et al. [10] study this problem under a revenue maximization objective, and develop provable approximation algorithms. Their results have been extended by Cao et al. [6] to the general case where demands for spectrum vary stochastically. In general, the bidders cannot be expected to be completely truthful, especially if they are allowed to trade their spectrum allocations in a tertiary market, raising the need for truth revealing market mechanisms. There is a lot of work on developing efficient auction mechanisms in such a setting, e.g., [12], [14].

Most of the work discussed above deals with the one-hop setting. Several papers have explored the end-to-end throughput capacity in cognitive networks, e.g., [15], [16], [3]. These papers give constant factor approximations to the total throughput possible between a set of connections, when nodes can opportunistically switch channels. Lin et al. [16] show that there is a significant gain in the throughput capacity by the use of opportunistic frequency assignment.

Our work is an extension of Steenstrup [23], who presents a greedy heuristic to compute a minimum such partition. However, this heuristic could be highly suboptimal in the worst case, as discussed in Figure 1, where we describe a family of feasible configurations on graphs for which no frequency is assigned by the heuristic. Steenstrup gives a distributed implementation of the heuristic [22].

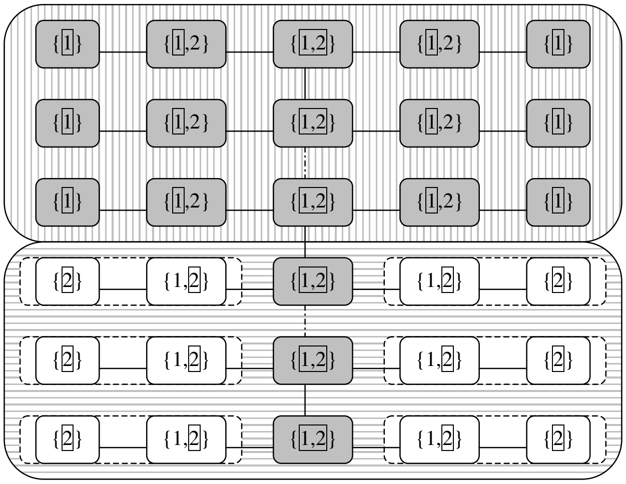

Fig. 2. Figure shows a tree with a feasible assignment $F_v^R, \forall v \in V$, shown inside each node over $\mathcal{F} = \{1, 2\}$. $F_v^T = \bigcap_{u \in N[v]} F_u^R$ is highlighted by a box. Note that $\forall \{u, v\} \in E, F_u^T \cap F_v^T \neq \emptyset$. Therefore, no frequencies are assigned by the heuristic in [23] even though F^R is a feasible assignment for the tree. Also, note that *greedy* choice for expanding clusters fails in keeping the size of the partition small. W.L.O.G., starting at a node containing only 1, the algorithm tries to extend the cluster containing the node as much as it can, greedily; the greedy choice makes the cluster highlighted as shaded nodes. This forces lots of small clusters (shown as dashed bubbles) for a total of $\frac{n}{5} + 1$ clusters. OPT on the other hand constructs a partition of size 2 shown by the two large bubbles with horizontal and vertical patterns.

III. Minimizing the Number of Clusters

A. CFRP *is NP-Complete*

In this section, we show that the decision version of CFRP (D-CFRP, in short) is NP-complete. The input to D-CFRP contains, additionally, a positive integer K and we are asked if there is a set of clusters of size at most K satisfying the constraints of the CFRP problem. The NP-completeness proof uses a reduction from the classical *dominating set* problem to D-CFRP problem. As a by-product of this proof we get a hardness of approximation proof for CFRP, showing essentially that CFRP is as hard to approximate as dominating set.

Given a graph $G = (V, E), D \subseteq V$ is a *dominating set* if for every $v \in V, \exists u \in D, d(u, v) \leq 1$. The *minimum dominating set* (MDS) problem takes as input a graph G and seeks to find a smallest dominating set of G. The decision version of MDS (DS, in short), additionally takes a positive integer K as input and asks if G has a dominating set of size at most K. It is well known that DS is NP-complete on many graph classes; furthermore, MDS is notoriously hard to approximate on general graphs [9].

Theorem 1: D-CFRP is NP-complete.

Proof: It is easy to see that the D-CFRP is in NP. To see that the problem is NP-hard, we give a polynomial time reduction from DS to D-CFRP. Define a ball of radius ℓ around v, $B(v,\ell) = \{u | d(v,u) \leq \ell\}$, to be the set of all vertices u within distance at most ℓ from v. Now, we construct an instance of CFRP from an instance of DS via the following procedure. Let $G = (V, E)$ be the graph in the instance of DS. In Lines (1)-(5), we construct a graph $H = (V', E')$, which consists of all of G plus, for each vertex v in G, a copy v', that is connected to v by an edge. In Lines (6)-(11), we assign the set of frequencies F_v^R to each vertex u in H using the rule that F_u^R is assigned the set of all labels of vertices in $B(u, 2)$.

```
DS2CFRP($G$)
1.    $(V', E') \leftarrow (\emptyset, \emptyset)$
2.    for each $v \in V(G)$ do
3.        $V' \leftarrow V' \cup \{v'\}$
4.        $E' \leftarrow E' \cup \{\{v, v'\}\}$
5.    $H \leftarrow (V(G) \cup V', E(G) \cup E')$
6.    for each $v \in V(H)$ do
7.        $F_v^R \leftarrow \emptyset$
8.    for each $v \in V(H)$ do
9.        for each $u \in B_H(v, 2)$ do
10.           $F_u^R \leftarrow F_u^R \cup \{v\}$
11.   return $H$
```

Next, we show that for the graph H there is a solution to the CRFP problem with K clusters if and only if G has a dominating set of size K.

"$\Leftarrow$": Suppose that the G has a dominating set of size K. Call this $D^* = \{d_1, d_2, \ldots, d_K\}$. Construct a partition $\mathcal{V} = \{V_1, V_2, \ldots, V_K\}$ of H as follows. Start with $V_i = \emptyset$ for each i.

- Add each vertex d_i to V_i.
- For each vertex $v \in V(G) \setminus D^*$, pick an arbitrary d_i that is adjacent to v and add v to V_i.
- For each vertex $v' \in V'$, add v' to V_i if its copy v is in V_i.

It is easy to see that $\mathcal{V}$ is a partition of $V(H)$ and each V_i is connected. Next, we show that $\bigcap_{v \in V_i} F_v^T \neq \emptyset, \forall i$. Observe that each V_i consists of some vertices in $B(d_i, 1)$ along with some vertices in V' that are adjacent to vertices in $B(d_i, 1)$. By the way the sets F_v^R are constructed, it is easy to see that d_i belongs to F_u^R for each $u \in V_i$ and furthermore d_i also belongs to F_u^R for each u that is adjacent to some vertex in V_i. This implies that $d_i \in F_u^T$ for all $u \in V_i$, thereby implying that $\bigcap_{v \in V_i} F_v^T \neq \emptyset, \forall i$.

"$\Rightarrow$": Suppose that $\mathcal{V} = \{V_1, V_2, \ldots, V_K\}$ is a partition of $V(H)$ that is a feasible solution to the CFRP problem. Let U_i denote $V_i \setminus V'$ for each i. Then $\{U_1, U_2, \ldots, U_K\}$ is a partition of $V(G)$. Suppose for each U_i, there is a v_i in $V(G)$ (not necessarily in U_i) such that $U_i \subseteq B(v_i, 1)$. Then, $\{v_1, v_2, \ldots, v_K\}$ is a dominating set of G.

So, suppose for the sake of contradiction, there is a $U_i \not\subset B(v, 1)$ for any $v \in V(G)$. Since V_i is a cluster that is part of a solution to CFRP and $U_i \subseteq V_i$, it must be the case that there is an $x \in F_v^T$ for all $v \in U_i$. By construction of the instance H, x is the label of a vertex in H and $V_i \subseteq B(x, 2)$. There are two cases, depending on whether (i) $x = v' \in V'$ or (ii) $x = v \in V(G)$.

(i) $x = v' \in V'$. In this case, $U_i \subseteq B(v, 1)$, contradicting our assumption that $U_i \not\subset N_G[v]$ for any $v \in V(G)$.

(ii) $x = v \in V(G)$. In this case also, $U_i \subseteq B(v, 1)$. Otherwise, if for some $u \in U_i$, $d_G(u, v) \geq 2$, then u has a neighbor in H, namely u', such that v does not belong to $F_{u'}^R$. This contradicts the fact that $v \in F_u^T$ for all $u \in U_i$.

■

B. Inapproximability of CFRP

Here, we show that unless NP $\subset$ DTIME$(n^{O(\log \log n)})$, the CFRP problem cannot be approximated within a ratio $(1 - \varepsilon) \ln n$, for any $\varepsilon > 0$. We borrow the reduction from the proof of Theorem 1 to show that DS is *L-reducible* to CFRP. Papadimitriou and Yannakakis [19] introduced the concept of L-reduction. It is defined in the following:

Definition 1 (L-reduction): Let A and B be two optimization problems. Let c_A and c_B be there corresponding cost functions. A is L-reducible to B if there are two constants $\alpha, \beta > 0$, such that:

1) there exists a polynomial time computable function which transforms an instance x of A into an instance x' of B such that:

$$\text{OPT}_B(x') \leq \alpha \cdot \text{OPT}_A(x),$$

2) there exists a polynomial time computable function which transforms any solution y' of x' into a solution y of x such that:

$$|c_A(x, y) - \text{OPT}_A(x)| \leq \beta \cdot |c_B(x', y') - \text{OPT}_B(x')|$$

For our purpose, the cost functions for DS and CFRP are the size of the dominating set and the size of the partition, respectively. We will describe a polynomial time computable function that maps an instance of DS to that of CFRP such that the optima are identical. Next, we will describe a polynomial time computable function that maps a feasible solution to an instance of CFRP to a feasible solution of DS such that the costs of the solutions are identical as well. This will satisfy the second condition of the definition of an L-reduction. Using this scheme, we will show that if there is a deterministic algorithm for CFRP that runs in polynomial time to produce a solution within $(1 - \varepsilon) \ln n$ of the optimal partition, for any $\varepsilon > 0$ then using the L-reduction, one can construct a deterministic algorithm for DS which runs in polynomial time and produces a solution within $(1 - o(1)) \ln n$ of the optimal dominating set. This, in turn will imply that all problems in NP can be solved in deterministic time $n^{O(\log \log n)}$, contradicting the widely held belief that no sub-exponential time exact algorithms exist for any problem in NPC.

Lemma 1: DS is L-reducible to CFRP.

Proof: To prove the L-reduction, we simply note that we can use the transformation procedure DS2CFRP(G) given in the proof of Theorem 1 exactly as it is to map an instance x of DS to an instance x' of CFRP; the "⇒" direction of the proof of Theorem 1 shows that $\text{OPT}_{\text{DS}}(x) = \text{OPT}_{\text{CFRP}}(x')$. This satisfies the first condition of an L-reduction. Also note that the "⇐" direction of the proof of Theorem 1 maps a feasible solution y' of an instance x' of CFRP to a feasible solution y of an instance x of DS. It is also easy to see that the second condition of an L-reduction is also satisfied,

$$\begin{aligned} c_{\text{DS}}(x, y) - \text{OPT}_{\text{DS}}(x) &= c_{\text{CFRP}}(x', y') - \text{OPT}_{\text{DS}}(x) \\ &= c_{\text{CFRP}}(x', y') - \text{OPT}_{\text{CFRP}}(x'), \end{aligned}$$

and the constants $\alpha = \beta = 1$. ■

Theorem 2: If NP $\not\subseteq$ DTIME$(n^{O(\log\log n)})$ then no polynomial time algorithm exists that can guarantee an approximation ratio of $(1 - \varepsilon)\ln n)$, for any $\varepsilon > 0$.

Proof: Suppose, for the sake of contraction, that there exists a polynomial time approximation algorithm for CFRP which achieves an approximation ratio of $(1 - o(1))\ln n$. Note that we showed in the proof of Lemma 1, a transformation of any instance x of DS to an instance x' of CFRP such that, $\text{OPT}_{\text{DS}}(x) = \text{OPT}_{\text{CFRP}}(x')$ and that the cost of any feasible solution y' of instance x' of CFRP is exactly equal to the cost of a feasible solution y to x of DS which maps to x'. So, we have the following:

$$\frac{c_{\text{DS}}(x, y)}{\text{OPT}_{\text{DS}}(x)} = \frac{c_{\text{CFRP}}(x', y')}{\text{OPT}_{\text{CFRP}}(x')} \leq (1 - o(1))\ln n$$

The last inequality then shows, by an extension to the result by Feige [9], that NP $\subset$ DTIME$(n^{O(\log\log n)})$. A contradiction. ■

IV. Other Greedy Heuristics for the CFRP problem

In this section, we explore another greedy objective that seeks to maximize the size of a cluster $S \subseteq V$ with the objective of minimizing the size of the fragmentation in $G \setminus G[S]$. This then leads to a recursive algorithm that greedily picks large subgraphs $G[S_i]$ which when removed from $G \setminus \bigcup_{j=1}^{i-1} G[S_j]$ minimizes the number of components. This algorithm is lead to by a simple observation that in any optimal partition $\mathcal{T} = \{T_1, T_2, \ldots, T_{K_{\text{OPT}}}\}$ of G, there exists a sequence $(T_1, T_2, \ldots, T_{K_{\text{OPT}}})$ such that for any $1 \leq i < K_{\text{OPT}}, G \setminus \bigcup_{j=1}^{i} G[T_j]$ is connected.

Theorem 3: Any optimal solution $\mathcal{T} = \{T_1, T_2, \ldots, T_{K_{\text{OPT}}}\}$ of the CFRP problem has at least 2 clusters whose removal leaves the rest of the graph connected.

Proof: Let $\mathcal{T} = \{T_i | 1 \leq i \leq K_{\text{OPT}}\}$ denote an optimal partition of G. Construct a simple cluster graph $G[\mathcal{T}] = (V', E')$ thus: let $V' = \{T_i\}$ and $E' = \{\{T_i, T_j\} | u \in T_i, v \in T_j, \{u, v\} \in E\}$. It is easy to see that $G[\mathcal{T}]$ is connected. It then follows from a well know observation that any n vertex connected graph has at most $n - 2$ cut vertices. This implies that there are at least 2 vertices in $G[\mathcal{T}]$ whose deletion leaves the rest of the graph connected. This then implies a natural ordering of $\mathcal{T} = (T_1, T_2, \ldots, T_{K_{\text{OPT}}})$. ■

A. Greedy Approach

But, how do we find T_1? Since the CFRP problem is NP-hard, it is unlikely that there exists a polynomial time algorithm that finds T_1. We consider the following greedy algorithm based on the above discussion.

```
GREEDYC($G$)
1.    $\mathcal{S} \leftarrow \emptyset$
2.    for each $v \in V$ do
2.1          $S_v \leftarrow$ largest cluster containing $v$ that
                  keeps the rest of the graph connected
3.    Let $S_u$ be a largest cluster. $G \leftarrow G \setminus G[S_u]$
4.    $\mathcal{S} \leftarrow \mathcal{S} \cup \{S_u\}$
5.    if $G = \emptyset$ then output $\mathcal{S}$
6.    else goto "Step 2."
```

Note that applying GREEDYC to the example in Figure 2 yields an optimal solution. Unfortunately, as shown in the next section, it is easy to see that GREEDYC yields at least a factor $\Omega(n^{\varepsilon})$ more clusters than an optimal solution.

B. $\Omega(n^{\varepsilon})$ performance for GREEDYC

In this section we provide a simple example for which GREEDYC produces at least $\Omega(n^{\varepsilon})$ clusters whereas an optimal solution would yield $k + 2$ clusters, for any $0 < \varepsilon < 1$ and any $k \geq 2$.

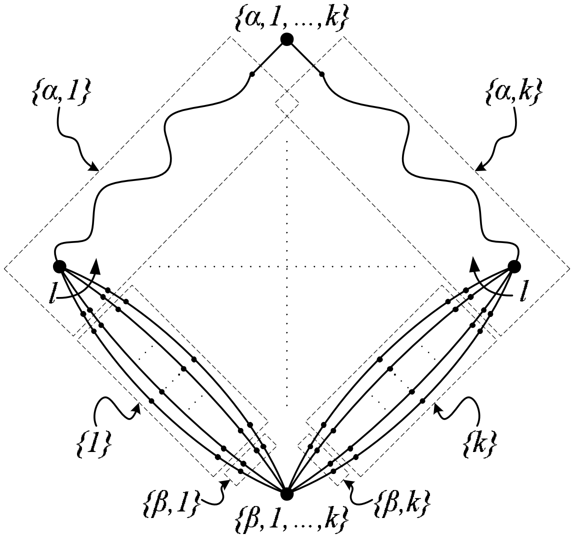

Fig. 3. Example showing $\Omega(n^{\varepsilon})$ performance ratio for GREEDYC. The boxes represent vertices having common labels (values for F^R). The values are indicated by sets pointing to the boxes. The labels for the top-most and bottom-most vertices appear above and below the respective vertices. In this example, OPT has a partition of size $k + 2$ whereas GREEDYC constructs a partition of size $kl + 2$ for appropriate choice of the numbers of vertices in each "box".

The Construction: We present a construction of a family of planar graphs for $0 < \varepsilon < 1$. The graph consists of k identical structures called "brooms". Each broom has l "bristles". Each bristle is a path containing p vertices. The broom consists of a "broom-stick" which is a path containing more than lp vertices. For each vertex v on the broom-stick, $F_v^R = \{\alpha, i\}$. Distinguish the two ends of the stick as v' and v''. Next, for each internal vertex u in each bristle, $F_u^R = \{i\}$; the two "tips" of each bristle u' and u'' have values $\{\alpha, i\}$ and $\{\beta, i\}$, respectively. Next, add an edge $\{v', u'\}$, for u' on each bristle. Next, add two vertices, A and B, and let $F_A^R = \{\alpha, 1, 2, \ldots, k\}$ and $F_B^R = \{\beta, 1, 2, \ldots, k\}$. Add $\{A, v''\}$ to the edge set for each of the k brooms. Add $\{u'', B\}$ for each of the l bristles u'' and for each of the k brooms to finish the construction of a planar graph. It is easy to see that the F^R assignment is feasible.

To see the performance of OPT, note that each of the k constituent brooms forms a block in the partition having $F_v^T = \{i\}$. The remaining two vertices, A, B, having F^T values $\{\alpha\}$ and $\{\beta\}$, respectively, for a partition of size $k+2$. GREEDYC, on the other hand, never chooses a broom since it greedily chooses A and all of the broom-sticks instead, since more than half the vertices fall in this connected component, assigning $F^T = \{\alpha\}$. Following this first greedy choice, all remaining choices are bad as then each of the lk bristles end up getting chosen just to maintain feasibility, for a total of $lk + 2$ blocks in the partition. This observation immediately leads to the following:

Theorem 4: GREEDYC is a $\Omega(n^\varepsilon)$ approximation to the CFRP problem for any $0 < \varepsilon < 1$ and any $k \geq 2$.

V. Optimal solutions for Trees

Let $T = (V, E)$ be the given tree. Pick an arbitrary $r \in V$ and root T at r. For any $v \in V$ and frequency $f \in F_v^T$, let OPT(v, f) denote the smallest number of clusters in a feasible solution to CFRP on the subtree of T rooted at v with the restriction that the cluster C containing v can use frequency f, i.e., for all $u \in C$, $f \in F_u^T$. Let $S(v, f)$ denote a solution associated with OPT(v, f). Let OPT$(v) = \min_f$ OPT(v, f). Then, OPT(r) denotes the number of clusters in an optimal solution of CFRP on T. There is a simple dynamic programming algorithm for computing OPT(v, f) and $S(v, f)$. Process the children of v in some arbitrary order and for each child u and frequency $f' \in F_u^T$, compute OPT(u, f') and the associated solution $S(u, f')$. Let $f^* \in F_u^T$ be such that OPT$(u) =$ OPT(u, f^*). If OPT$(u, f) \leq$ OPT(u, f^*), then $S_u \leftarrow S(u, f)$ and "mark" u; otherwise $S_u \leftarrow S(u, f^*)$ (and do not "mark" u). To obtain a solution for the tree rooted at v, take $\cup_u S_u$ (the union taken over all children u of v), merge all the clusters containing marked children, and then add v to this merged cluster.

VI. A Relaxation of CFRP

In this section, we present a relaxation of the CFRP problem. Specifically, we relax the constraint that the clusters are required to form a partition of the vertex set. Instead we allow clusters to "overlap," i.e., have common vertices, but attempt to simultaneously minimize the amount of overlap as well as the number of clusters. We describe two approximation algorithms for this relaxed CFRP problem. The first algorithm is based on solving a linear program (LP) and then employing randomized rounding to obtain an integral solution. The second algorithm uses a greedy technique augmented by Lagrangian relaxation. This latter algorithm is quite easy to implement and this is what we use to produce the simulation results described in Section VII. Roughly speaking, both algorithms produce as output a collection $\mathcal{C}$ of clusters that covers the graph and has two additional properties: (i) the number of clusters in $\mathcal{C}$ is at most $O(\log n)$ times the number of clusters in an optimal solution to CFRP and (ii) each vertex, on average, appears in $O(\log n)$ clusters in $\mathcal{C}$.

A. LP-Rounding for a Relaxed Version of CFRP

For any collection $\mathcal{C}$ of node subsets, we define the *overlap* of a node v as the number of subsets in $\mathcal{C}$ that v belongs to. The *overlap* of the collection itself is the sum of the overlaps of all the nodes. We relax CFRP to allow clusters to overlap, while attempting to minimize the overlap of the collection of clusters. The overlap of a collection of clusters can be expressed as the sum of cardinalities of the clusters and thus the objective of our optimization problem, which we call R-CFRP (short for relaxed CFRP), is to pick a set of clusters whose sum of cardinalities is minimum. The constraints of R-CFRP are that (i) each node appears in at least one cluster and (ii) the total number of clusters is bounded above by K, for some integer input K. We express R-CFRP as the integer program (IP) given below. Let S be the set of all possible clusters. For each cluster $A \in S$, let x_A be a binary indicator variable denoting the inclusion of A in the solution. Denote by $S(v)$ the set of clusters that contain v.

IP: $$\min \sum_{A \in S} |A| \cdot x_A$$

subject to:

$$\begin{array}{cccr} \displaystyle\sum_{A \in S(v)} x_A & \geq & 1, & \forall v \in V \\ \displaystyle\sum_{A \in S} x_A & \leq & K & \\ x_A & \in & \{0, 1\}, & \forall A \end{array}$$

We obtain the natural LP-relaxation of the above IP by replacing each constraint $x_A \in \{0, 1\}$ by $x_A \geq 0$. This linear program (LP) is similar to the linear programs for the *metric k-median problem* [13] and the *clustering for min-sum cluster diameters problem* [7].

1) The Algorithm: Our *LP-rounding* algorithm proceeds in two stages:

1) Solve the LP-relaxation in the previous section to obtain x_A^* for each $A \in S$.
2) Independently, add each A to the solution with probability $= y_A^* = \min\{2 \ln n \cdot x_A^*, 1\}$.

Note that even though the LP has exponentially many variables, it has $O(n)$ constraints, and each variable appears in $O(n)$ constraints. Since it is a mixed packing and covering program with no negative constraints, the result of Young [27] gives an approximate solution to it in polynomial time.

2) Analysis: In this section, we present a simple analysis of LP-rounding algorithm. A solution to R-CFRP is said to be *t-feasible* if each vertex belongs to at least one cluster and the total number of clusters is at most $K \times t$. Thus a t-feasible solution satisfies the first set of constraints in IP exactly, while violating the second constraint by a factor of t.

Lemma 2: The solution to R-CFRP produced by the LP-rounding algorithm is $4 \ln n$-feasible with high probability, i.e., at least $1 - 2/n$.

Proof: We first bound by $1/n$ the probability that there exists a node that does not belong to any cluster. Fix a node v. If for any $A \in S(v)$, the probability $y^*_A = 1$, then v is covered by at least one cluster with probability 1. So we assume that for all $A \in S(v)$, $y^*_A < 1$. This implies that $y^*_A = 2 \ln n \cdot x^*_A$ for all $A \in S(v)$. In this case, we calculate an upper bound on the probability that v is not covered by any cluster, as follows.

$$\begin{aligned}
\mathbf{Pr}\left[\substack{v \text{ is not}\\ \text{covered by}\\ \text{any } A}\right] &= \prod_{A \in S(v)} (1 - y^*_A)\\
&\leq \prod_{A \in S(v)} \exp(-y^*_A)\\
&= \exp\left(-\textstyle\sum_A y^*_A\right)\\
&= \exp\left(-2 \ln n \textstyle\sum_A x^*_A\right)\\
&\leq \tfrac{1}{n^2}
\end{aligned}$$

The last inequality is from the fact that $\sum_A x^*_A \geq 1$. Using the union bound, we get that,

$$\mathbf{Pr}\left[\substack{\text{some } v \text{ is not}\\ \text{covered by any } A}\right] \leq \frac{1}{n}$$

Next, we bound by $1/n$ the probability that the number of clusters in the solution exceeds $K \cdot 4 \ln n$.

$$\begin{aligned}
\mathbf{E}\left[\text{number of clusters}\right] &= \sum_{A \in S} \mathbf{Pr}\left[A \text{ is chosen}\right]\\
&= \sum_{A \in S} y^*_A\\
&\leq 2 \ln n \sum_{A \in S} x^*_A\\
&\leq 2K \cdot \ln n
\end{aligned}$$

Since each A is chosen independently, using Chernoff bounds, we get:

$$\mathbf{Pr}\left[\substack{\#\text{ of clusters}\\ > 4K \cdot \ln n}\right] \leq \frac{1}{n}$$

The two probability bounds together imply that with probability at least $1 - 2/n$, every node is covered by at at least one cluster and the number of clusters is at most $K \cdot 4 \ln n$. ■

We now bound the probability that the overlap of the clusters chosen by the LP-rounding algorithm is high. Let OPT denote the optimal overlap of a solution to R-CFRP; in other words, OPT is the optimal value of the objective function of the IP.

Lemma 3: Let T denote the overlap of the solution to R-CFRP produced by the LP-rounding algorithm. Then,

$$\mathbf{Pr}[T > 4 \ln n \cdot \text{ OPT}] \leq \frac{1}{2}.$$

Proof:

$$\begin{aligned}
\mathbf{E}\left[T\right] &= \sum_{A \in S} |A| y^*_A\\
&\leq \sum_{A \in S} |A| 2 \ln n \cdot x^*_A\\
&\leq 2 \ln n \cdot \text{ OPT}
\end{aligned}$$

The last inequality simply follows from the fact that $\sum_{A \in S} |A| \cdot x^*_A$, the optimal value of the objective function of the LP-relaxation, cannot be larger than the optimal value of the objective function of the IP

Using Markov's inequality, we get:

$$\mathbf{Pr}\left[T > 4 \ln n \cdot \text{ OPT}\right] \leq \frac{1}{2}$$

■

To keep both the amount of overlap as well as the number of clusters small, we use the LP-rounding algorithm as follows. Let K_{OPT} be the number of clusters in an optimal solution to CFRP. Note that if $K \geq K_{\text{OPT}}$ in R-CFRP, then OPT is n, since each node in a feasible solution to CFRP appears in a single cluster. Hence, for such a value of K, the overlap of the solution produced by the LP-rounding algorithm is at most $4 \ln n \cdot n$ with probability at least 1/2. Repeating Stage 2 of the LP-rounding algorithm $O(\log n)$ times for the same value of K and outputting a solution with smallest overlap selected from among all of the $O(\log n)$ trials guarantees a solution with overlap at most $4 \ln n \cdot n$, with high probability, i.e., $1 - 1/n$. A "good" value for K can then be found as follows. For each candidate value of K, considered in the order $1, 2, 3, \ldots, n$, run the LP-rounding algorithm with $O(\log n)$ repetitions. Stop as soon as we reach a value of K for which find a solution with overlap at most $4 \ln n \cdot n$. With high probability, this algorithm will stop with $K \leq K_{\text{OPT}}$, yielding the following theorem.

Theorem 5: There is a randomized algorithm that in time polynomial in the size of the input produces, with high probability, a collection of clusters that cover the graph and satisfy the properties (i) each node appears in at most $4 \log n$ clusters on average and (ii) the number of clusters is at most $K_{\text{OPT}} \cdot \log n$.

B. Greedy Algorithm

We can obtain a greedy algorithm for R-CFRP by noting that R-CFRP can be expressed as a *set cover* problem in which the cost of each set is its cardinality. In such a setting, the natural greedy algorithm will repeatedly pick a set that covers the most as yet uncovered nodes per unit cost. This "greedy choice" appears in Step 3.1 below. A variety of different techniques for analyzing such greedy algorithms appear in Vazirani's book [24].

```
GREEDY1($G$)
for each $v \in V$ do $c_v \leftarrow 0$
$i \leftarrow 1; \mathcal{S} \leftarrow \emptyset; C \leftarrow \emptyset$
while $V$ is not covered do
    $S_i \leftarrow \text{argmin} \frac{|S_j|}{|S_j \setminus C|}$
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{S_i\}; C \leftarrow \bigcup_{j=1}^{i} S_j; i \leftarrow i+1$
```

The performance of GREEDY1 is quite poor. This can be seen by noting that the ratio $|S_j|/|S_j \setminus C|$ has a minimum value of 1 and this value is achieved by any set S_j that only covers new elements. Thus, as stated, the algorithm repeatedly picks an arbitrary cluster that only covers as yet uncovered nodes. This greedy algorithm has already been considered by Steenstrup [23] and has poor performance with regards to the number of clusters it uses (see Figure 2).

We modify GREEDY1 by looking for a subset S_j that minimizes $\frac{S_j + \lambda}{|S_j \setminus C|}$. Here $\lambda > 0$ is a Lagrangian multiplier that is assigned the value $\frac{n}{K_{\text{OPT}}}$, where K_{OPT} is the number of clusters in an optimal solution to CFRP. Since we do not know K_{OPT} to start with, we simply consider all possible values $1, 2, 3, \ldots, n$ for K_{OPT} and use the corresponding λ to determine the desirability of each set. We return the collection of sets with smallest overlap, taken over all possible choices of λ. We will refer to this modified algorithm as GREEDY2.

1) Analysis: Let $\mathcal{T} = \{T_1, T_2, \ldots, T_{K_{\text{OPT}}}\}$ denote an optimal solution to the CFRP problem. Note that this is a partition and therefore has overlap n. For any given family $\mathcal{Y}$ of K sets, define $\Phi(\mathcal{Y}) = \sum_i |Y_i| + \lambda K$, where λ is the Lagrangian multiplier. If $\lambda = \frac{n}{K_{\text{OPT}}}$ then $\Phi(\mathcal{T}) = \sum_i |T_i| + \lambda K_{\text{OPT}} = 2n$. Using analysis similar to the analysis of the greedy set cover ([24, p. 16]), it is not hard to show that $\Phi(\mathcal{S}) \leq \Phi(\mathcal{T}) \cdot \Theta(\log n)$. Since $\Phi(\mathcal{S}) = \sum_i |S_i| + \lambda K$ (by definition) and since $\Phi(\mathcal{T}) = 2n$, we obtain the following inequalities:

(i) $\sum_i |S_i| \leq 2n \cdot \Theta(\log n)$.

(ii) $\lambda K \leq 2n \cdot \Theta(\log n)$.

Inequality (i) states that the overlap of the collection $\mathcal{S}$ returned by GREEDY2 is $2n \cdot \Theta(\log n)$. Substituting $\lambda = n/K_{\text{OPT}}$ in Inequality (ii) yields $K \leq \Theta(\log n) \cdot K_{\text{OPT}}$. The conclusion of this analysis is stated in the following theorem.

Theorem 6: Algorithm GREEDY2 yields a collection $\mathcal{S}$ of at most $\Theta(\log n) \cdot K_{\text{OPT}}$ sets that cover the graph and have average overlap $\Theta(\log n)$.

VII. Simulation Results

We implemented a greedy algorithm that seeks to remove a largest feasible connected component (discussed in the caption of Figure 2), called GREEDY, and GREEDY2 as discussed earlier. For the simulations, we ran the two algorithms on randomly generated instances of the clustering problem on randomly generated *unit disk graphs* (UDGs). An n-vertex graph $G = (V, E)$ is said to be a UDG if there exists a map $\Phi : V \mapsto \mathbb{R}^2$ such that $\{u, v\} \in E$ if and only if $\|\Phi(u) - \Phi(v)\|_2 \leq 1$. UDGs provide a convenient model for homogeneous wireless networks.

Summary: We summarize the simulation setup and results here. We generated feasible instances of CFRP on random UDGs of three different types (shown in Figures 4,5, and 6) and ran GREEDY and GREEDY2 on them. Simulation results show that GREEDY, on average, performs well as compared to its worst case performance shown in Figure 2. However, GREEDY2 performs even better than GREEDY. For Type II graphs, the increase in performance ranges from about a factor of 30 (for a 60000 vertex UDG) to about 5 (for a 6000 vertex UDG). For Type III, we see a performance increase by about a factor of 5. For Type I, we observe only a modest increase in performance of about a factor 1.5. In the following, we provide more detail on how the instances were generated and what results were obtained. Following this, we discuss these results and give our speculation gleaned from our simulations.

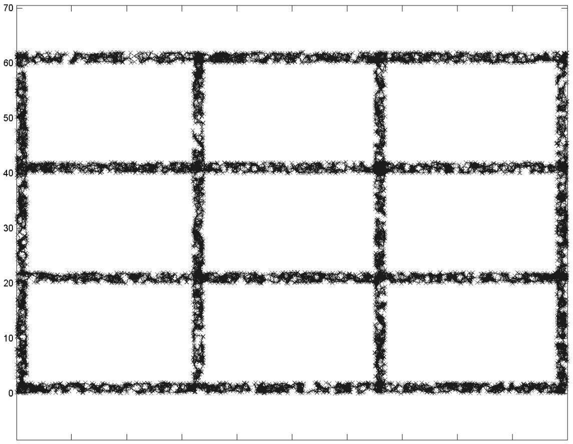

Fig. 4. Type - I

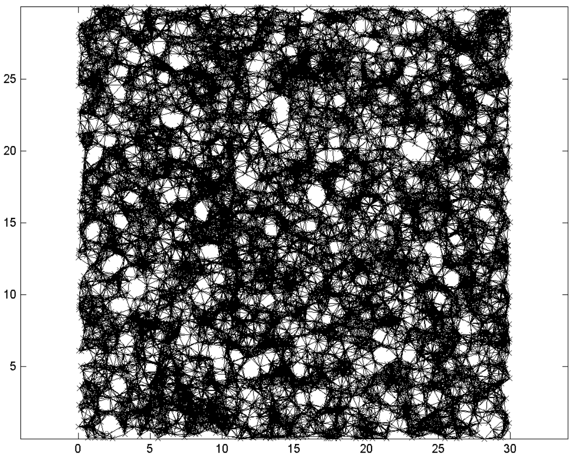

Fig. 5. Type - II

Simulation Setup and Results: We generated random UDGs from a choice of types that we describe later. We used uniform random distribution of a specified number of points within a specified rectangle as a basis for each type. One or more of such randomly generated set of points within their corresponding rectangles were then assembled to form the set of points for a UDG. Edges were then added between

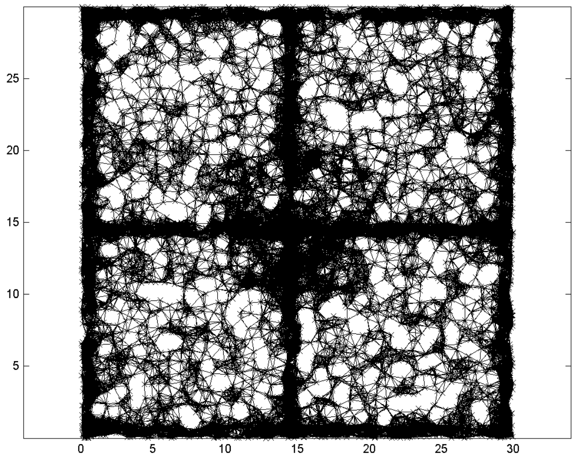

Fig. 6. Type - III

pairs of points that are within unit distance from each other. In all our simulations, we considered only connected UDGs. For each such randomly generated graph, we specified a frequency spectrum which was modeled as a contiguous set of integers. Initially, the entire spectrum would be made available to each vertex. For each graph, we also specified a number of *primary users*. These primary users' positions were chosen in two different ways: uniformly at random from amongst the UDG vertices, and uniformly at random from the bounding-box of the UDG. Each primary user chose a frequency from the spectrum uniformly at random. We also specified the *radius of influence* that applied to all the primary users. For any primary user, its choice of frequency was removed from the set of available frequencies of all the UDG vertices that appeared in its radius of influence. We considered only feasible configurations in which the set of available *transmission frequencies* at each UDG vertex would be non-empty. We then ran GREEDY and GREEDY2 on the random UDGs for the various types. We considered three types of random UDGs: (i) grid-like distribution (Type - I), (ii) uniform random distribution (Type - II), and (iii) combination (Type - III). Type - I consisted of a collection of thin, long rectangles that each contain points distributed uniformly at random within them to induce a connected subgraph. These rectangles were then arranged in equally spaced and intersecting rows and columns to form a connected UDG (Figure 4). Type - II consisted of a single rectangle with a specified number of points distributed uniformly at random to form a connected UDG (Figure 5). Type - III is a combination of Type - I and II (Figure 6). Table I contains average results from the simulations.

For our second set of simulations, we picked points uniformly at random from within the bounding box of the graph instead of choosing the positions from amongst those of the UDG vertices. We compare GREEDY and GREEDY2 for Type - I and III. Table II shows the results of this set of simulations.

For simulations of GREEDY2, we did not guess all possible values of K to seek K_{OPT}. Instead, we considered a geometrically increasing sequence of $K = 1, 2, 4, \ldots n$ to compute λ. The number of clusters then is within a factor of 2 of the guarantees of Theorem 6.

a) Discussion:: As argued earlier, removal of a cluster that causes a large fragmentation leads to a poor clustering. However, by allowing clusters to overlap prevents fragmentation. Furthermore, covering a large number of uncovered vertices using small sets attempts to minimize the overlap between clusters. Simulation results show that the average number of overlap of cluster appears to be quite small. We feel that in the case of UDGs, this average overlap is bounded above by a constant. This would then suggest that for wireless networks networks that are homogeneous, it is not necessary that these clusters partition the network. Observe that the primary motivation, as described by Steenstrup [23], [22] for minimizing the number of clusters is to try to minimize the overhead of vertices switching frequencies while maintaining network connectivity; if a vertex belongs to multiple clusters then the switching overhead at any node will be proportional to the number of clusters that contain it. However, if this overlap is at most a constant then this overhead will be independent of the network size. This would imply that by relaxing the constraint of obtaining a network partition, we introduce bounded switching overhead while simultaneously keeping the number of clusters small.

ACKNOWLEDGMENTS

The work of V.S. Anil Kumar and Madhav Marathe has been partially supported by NSF Nets Grant CNS-062694, HSD Grant SES-0729441, CDC Center of Excellence in Public Health Informatics Grant 2506055-01, NIH-NIGMS MIDAS project 5 U01 GM070694-05, and DTRA CNIMS Grant HDTRA1-07-C-0113. We thank the following people for their generosity. In a very short amount of time, they made their computers available for simulation purposes for long hours. We thank: Steve Beck, Rajankumar Bhatt, Kim Farrell, Mahdiar Hariri, Ross Johnson, Lindsey Knake, Qinghua Liu, Tim Marler, Anith Mathai, Chris Murphy, Amos Patrick, Molly Patrick, Brent Rochambeau, Brian Smith, Uday Verma, Jingzhou Yang. We thank SANTOS$^{\text{TM}}$for tolerating every time we stepped on his toes. Finally, we thank the anonymous referees for helping improve the paper.

REFERENCES

[1] I. Akyildiz, W. Lee, M. Vuran, and S. Mohanty. Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey. *Computer Networks*, 50:2127–2159, 2006.

[2] I. Akyildiz, W. Lee, M. Vuran, and S. Mohanty. Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey. *Computer Networks Journal*, 2006.

[3] Mansoor Alicherry, Randeep Bhatia, and Li (Erran) Li. Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks. In *MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking*, pages 58–72, New York, NY, USA, 2005. ACM Press.

[4] M. Buddhikot and K. Ryan. Spectrum management in coordinated dynamic spectrum access based cellular networks. In *IEEE DySPAN*, 2005.

[5] L. Cao and H. Zheng. Spectrum allocation in ad hoc networks via local bargaining. In *SECON*, 2005.

Type	n	trials	$\|\mathcal{F}\|$	$\|P\|$	radius	GREEDY	GREEDY2	max. overlap	average overlap
I	6,400	7	100	100	10.0	98.8	65.7	4.8	1.8
II	60,000	5	100	100	10.0	611.6	19.2	17.4	10.3
II	10,000	5	100	100	10.0	205	31.2	15.4	6.2
II	6,000	6	88	30	5.0	51.2	9.5	8.3	5.3
III	12,400	5	90	50	10.0	126.8	26.4	17	6.3
III	6 ,400	4	100	30	5.0	46	9.5	8.8	5.4
III	6 ,000	4	100	30	5.0	46	10	9	5.7

TABLE I
Simulation results for the case when primary user positions were chosen from amongst the UDG vertices. First column shows the type, second column contains the number of vertices, third column gives the number of independent random trials, fourth column shows the number of frequencies, fifth column shows the number of primary users, sixth column contains the radius of influence for each primary user, seventh column contains the average number of clusters for GREEDY, eighth column contains the average number of clusters for GREEDY2, the nineth and tenth columns contain the average maximum cluster overlap at any vertex, and average of the average cluster overlap at a vertex, respectively.

Type	n	trials	$\|\mathcal{F}\|$	$\|P\|$	radius	GREEDY	GREEDY2	max. overlap	average overlap
I	6,400	7	100	100	10.0	80.5	49.1	4.8	1.9
III	6,400	4	100	30	5.0	50.2	12.2	9.5	5.8
III	6,000	4	100	30	5.0	45	9.5	8.5	5.3

TABLE II
Simulation results for the case when primary user positions were picked uniformly at random from within the bounding box of the graph. First column shows the type, second column contains the number of vertices, third column gives the number of independent random trials, fourth column shows the number of frequencies, fifth column shows the number of primary users, sixth column contains the radius of influence for each primary user, seventh column contains the average number of clusters for GREEDY, eighth column contains the average number of clusters for GREEDY2, the nineth and tenth columns contain the average maximum cluster overlap at any vertex, and average of the average cluster overlap at a vertex, respectively.

[6] L. Cao and H. Zheng. Sparta: Stable and efficient spectrum access in next generation dynamic spectrum networks. In *INFOCOM*, 2008.

[7] Moses Charikar and Rina Panigrahy. Clustering to minimize the sum of cluster diameters. *J. Comput. Syst. Sci.*, 68(2):417–441, 2004.

[8] FCC. Unlicensed operation in the tv broadcast bands notice of proposed rulemaking (nprm). In *ET Docket No. 04-186*, 2004.

[9] Uriel Feige. A threshold of ln n for approximating set cover. *J. ACM*, 45(4):634–652, 1998.

[10] Sorabh Gandhi, Chiranjeeb Buragohain, Lili Cao, Haitao Zheng, and Subhash Suri. A general framework for clearing auction of wireless spectrum. In *Proceedings of the IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, pages 22–33, 2007.

[11] A. Ghasemi and E Sousa. Spectrum sensing in cognitive radio networks: the cooperation-processing tradeoff. *Wireless Communications and Mobile Computing*, 7(9):104–1060, 2007.

[12] J. Huang, R. Berry, and M. Honig. Auction-based spectrum sharing. In *WiOpt*, 2004.

[13] Kamal Jain and Vijay V. Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and Lagrangian relaxation. *J. ACM*, 48(2):274–296, 2001.

[14] Z. Ji and K. Liu. Dynamic spectrum sharing: A game theoretical overview. *IEEE Communications Magazine*, pages 88–94, 2007.

[15] Murali Kodialam and Thyaga Nandagopal. Characterizing the capacity region in multi-radio multi-channel wireless mesh networks. In *MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking*, pages 73–87, New York, NY, USA, 2005. ACM Press.

[16] X. Lin and S. Rasool. A distributed and provably efficient joint channel-assignment, scheduling and routing algorithm for multi-channel multi-radio wireless mesh networks. In *Proc. of IEEE INFOCOM*, 2007.

[17] J. Neel and J. Reed. Performance of distributed dynamic frequency selection schemes for interference reducing networks. In *Milcom*, 2006.

[18] N. Nie and C. Comaniciu. Adaptive channel allocation spectrum etiquette for cognitive radio networks. In *IEEE DySPAN*, 2005.

[19] Christos H. Papadimitriou and Mihalis Yannakakis. Optimization, approximation, and complexity classes. *J. Comput. Syst. Sci.*, 43(3):425–440, 1991.

[20] C. Peng, H. Zheng, and B. Zhao. Utilization and fairness in spectrum assignemnt for opportunistic spectrum access. *Mobile Networks and Applications (MONET)*, 11:555–576, 2006.

[21] S. Ramanathan. A unified framework and algorithm for channel assignment in wireless networks. *Wireless Networks*, 3:71–82, 1997.

[22] Martha E. Steenstrup. Channel selection among frequency-agile nodes in multihop wireless networks. In *ADHOC '05: Proceedings of the 5th Scandinavian Workshop on Wireless Ad-hoc Networks*, Stockholm, Sweden, 2005. Wireless@KTH.

[23] Martha E. Steenstrup. Opportunistic use of radio-frequency spectrum: A network perspective. In *DySPAN '05: Proceedings of the 1st IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks*, pages 638–641, Washington, DC, USA, 2005. IEEE Computer Society.

[24] Vijay V. Vazirani. *Approximation Algorithms*. Springer-Verlag New York, Inc., New York, NY, USA, 2001.

[25] B. Wild and K. Ramchandran. Detecting primary receivers for cognitive radio applications. In *Proceedings of the IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, pages 124–130, 2005.

[26] Y. Xing, R. Chandramouli, S. Mangold, and S. Shankar. Dynamic spectrum access in open spectrum wireless networks. *IEEE Journal on Selected Areas in Comm*, 24, 2006.

[27] Neal E. Young. Sequential and parallel algorithms for mixed packing and covering. In *The 42nd Annual Symposium on Foundations of Computer Science (IEEE FOCS)*, pages 538–546, 2001.