

A Greedy Asynchronous Distributed Interference Avoidance Algorithm

Behtash Babadi and Vahid Tarokh

School of Engineering and Applied Sciences
Harvard University

Outline of Topics

- 1 Introduction
- 2 The GADIA Algorithm
- 3 Performance Analysis
 - Static Case
 - Time-varying Case
 - Spectrum Sensing Time
 - Adaptive Rate
- 4 Conclusions

Outline

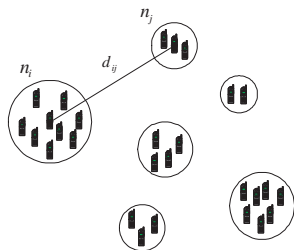
- 1 Introduction
- 2 The GADIA Algorithm
- 3 Performance Analysis
 - Static Case
 - Time-varying Case
 - Spectrum Sensing Time
 - Adaptive Rate
- 4 Conclusions

Introduction

- In many emerging wireless networks no central frequency allocation authority is naturally available.
- Examples are
 - ▶ Ad hoc Networks
 - ▶ Cognitive Radios
- Optimal frequency allocation requires full knowledge of the spatial distribution profile of the network nodes.
- This makes distributed frequency allocation an important but mostly uncharted territory in wireless networking.
- Objective: Dynamic assignment of the frequency bands to the users in the network in order to minimize the interference.

System Model

- Various networks are naturally clustered, *i.e.* the network elements are partitioned into a union of clusters.
- Examples include combat scenarios, WLAN hot spots, WLAN, etc.
- N clusters, $c_i, i = 1, \dots, N$, where each cluster has a cluster head responsible for managing some of the network functions. d_{ij} denotes the distance between the cluster heads of c_i and c_j .



Interference Model

- We assume that the updates are taking place at times t_1, t_2, \dots .
- The interference experienced by c_i caused by all the other clusters is

$$I_{c_i}(N, \{d_{ij}\}, l) = \sum_{j \neq i} \frac{KP_0}{d_{ij}^\eta} \delta(s_i(l), s_j(l))$$

where l denotes the update time t_l , $l = 1, 2, \dots$.

- The aggregate interference of the network at time l is

$$I(N, \{d_{ij}\}, l) = \sum_i I_{c_i}(N, \{d_{ij}\}, l) = \sum_i \sum_{j \neq i} \frac{KP_0}{d_{ij}^\eta} \delta(s_i(l), s_j(l))$$

- Note that this channel model is not necessary for the convergence of our algorithm (our algorithm works with any other channel model as long as it is reciprocal).

Similar Problems and Existing Approaches

- There are a number of proposed solutions to similar problems in different contexts (graph coloring, iterative waterfilling, etc.)
- These approaches have either of these drawbacks:
 - ▶ Excessively simplifying the interference models
 - ▶ Not fully decentralized
 - ▶ Require too much information exchange between autonomous nodes/clusters
 - ▶ Too complex to implement
 - ▶ Suffer from all the above shortcomings

Similar Problems and Existing Approaches (cont.)

- C. Peng, H. Zheng, and B. Y. Zhao [2006] propose that secondary users choose their spectrum according to their information about their local primary and secondary neighbors, in a cognitive network setting (Vertical sharing).
- Nodes are the vertices of a graph and any two interfering nodes are connected with an edge. This turns the problem into the graph multi-coloring problem.
- A sub-optimal solution to the graph multi-coloring, using an approximation algorithm to the graph labeling problem.
- Drawbacks:
 - ▶ Not fully decentralized
 - ▶ The interference model is excessively simplified
 - ▶ Too much message-passing among the nodes
 - ▶ High complexity

Similar Problems and Existing Approaches (cont.)

- Similar works in the context of Digital Subscriber Lines (DSL).
- W. Yu, G. Ginis, and J. M. Cioffi [2002] have proposed a method of iterative waterfilling in order to solve the problem of optimal PSD shaping in DSL applications.
- Each user must know a weighted sum of the PSD of the other users (interference), in order to do waterfilling.
- Drawbacks:
 - ▶ High computational complexity.
 - ▶ Nash equilibrium point does not necessarily correspond to the optimal answer.
 - ★ For instance, in a two-user scenario, if both users start with a flat PSD initially, iterative waterfilling does not change their PSD.
 - ★ This is clearly a Nash equilibrium point, but is far away from the optimal answer.

Similar Problems and Existing Approaches (cont.)

- R. Cendrillon, J. Huang, M. Chiang, and M. Moonen [2006], [2007] consider the problem for a DSL system with N users and K tones.
- The achievable bit-rate of user n is

$$R^n \triangleq \sum_{k=1}^K \log \left(1 + \frac{s_k^n}{\sum_{m \neq n} \alpha_k^{n,m} s_k^m + \sigma_k^n} \right)$$

where s_k^n is the transmission power of user n over tone k , $\alpha_k^{n,m}$ is the normalized cross-talk channel between users n and m and σ_k^n is the noise level of tone k for user n .

- The optimization problem is:

$$\max_{s_k^n, \forall n, k} \sum_i w_i R^i \quad s.t. \quad \sum_k s_k^n \leq P^n, \forall n$$

for a given set of $0 \leq w_1, \dots, w_N \leq 1$ such that $\sum_i w_i = 1$.

- This problem can be solved iteratively in a centralized fashion and converges to the optimal values.
- However, it is very complicated due to being centralized.

Similar Problems and Existing Approaches (cont.)

- Very hard to solve in a decentralized manner.
- The optimization problem is relaxed based on introducing a virtual user with fixed thresholds.
- The throughput of the virtual user (from the viewpoint of user n) is

$$R^{n,ref} \triangleq \sum_k \log \left(1 + \frac{\tilde{s}_k}{\tilde{\alpha}_k^n s_k^n + \tilde{\sigma}_k} \right)$$

where \tilde{s}_k is a fixed power assignment over tone k for the virtual user, $\tilde{\sigma}_k$ is the noise over tone k , and $\tilde{\alpha}_k^n$ is the cross-talk channel of user n and the virtual user over tone k .

- The relaxed optimization problem for each user n :

$$\max_{s_k^n \forall k, w^n} w^n R^n + (1 - w^n) R^{n,ref} \quad s.t. \quad \sum_k s_k^n \leq P^n$$

where the maximization is jointly over w^n and s_k^n , $k = 1, \dots, K$.

Similar Problems and Existing Approaches (cont.)

- Each user solves the relaxed optimization problem locally across different tones. The knowledge of a weighted sum of the PSD of the other users (interference) is required.
- The convergence is proved only in high SNR regime.
- The achievable region resulted by $\sum_i w_i R^i$ over all the values of $0 \leq w_i \leq 1, \forall i$ such that $\sum_i w_i = 1$ is close to the achievable region of the optimal centralized solution.
- No one-to-one correspondence between the points of the achievable regions of the optimal (centralized) and decentralized algorithms.
- The algorithm does not necessarily converge to optimal values.

Similar Problems and Existing Approaches (cont.)

- For the case of asynchronous transmission (in the presence of ICI), the optimization can not be separated across the tones. They have therefore used heuristic optimization approaches with no convergence guarantee.
- Drawbacks:
 - ▶ Simplified model for the coupling of the users
 - ▶ Stringent constraints for the uniqueness of the Nash equilibrium point
 - ▶ The convergence is only proved in high SNR regime
 - ▶ No guarantee on the optimality

Contributions of Our Work

- Our proposed dynamic frequency allocation algorithm is fully distributed
 - ▶ No information exchange between autonomous devices is needed
 - ▶ No knowledge of the existence of other autonomous entities is required
- The proposed algorithm is simple and has low computational complexity.
- It can be used in conjunction with any realistic wireless radio channel model such as those commonly employed in wireless standards (e.g., Hata model, Okumura model, etc.)
- Convergence of this algorithm to a sub-optimal solution is proved
- We have established performance bounds showing that this sub-optimal solution is near-optimal under various practical node activity models.
- The algorithm achieves at least 90% of the Shannon capacities corresponding to the optimum centralized frequency band assignment, even in presence of time-varying activity of clusters.

Assumptions

- At each time slot for any cluster *at most* one user is transmitting and one user is receiving.
 - ▶ Alternative scenarios are possible, e.g., users transmit and receive through the cluster head.
- The distances between clusters are much larger than the size of clusters and bounded below by a distance δ .
- The rate of change of the spatial distributions of the clusters in the network and the underlying channels is much less than the processing/transmission rate.

Assumptions (cont.)

- Each user transmits with power KP_0 , where K is a function of frequency.
 - ▶ This assumption can be relaxed.
- Path loss with exponent η . No shadowing and fading is assumed.
- r different accessible transmission bands, b_1, \dots, b_r .
- At time t , the i th cluster is in state $s_i(t) \in \{1, 2, \dots, r\}$, corresponding to the index of the transmission band it is using.
- Performance metric: Aggregate interference of the network.

Outline

1 Introduction

2 The GADIA Algorithm

3 Performance Analysis

- Static Case
- Time-varying Case
- Spectrum Sensing Time
- Adaptive Rate

4 Conclusions

GADIA Algorithm

GADIA Algorithm: *Clusters scan all the frequency bands b_1, \dots, b_r in an asynchronous manner over time. Each cluster chooses the frequency band in which it experiences the least aggregate interference from other clusters.*

- The cluster head scans all the frequency bands and estimates/measures the interference it experiences in each frequency band.
- The cluster head chooses the new transmission frequency band.

An Example of The Update Process

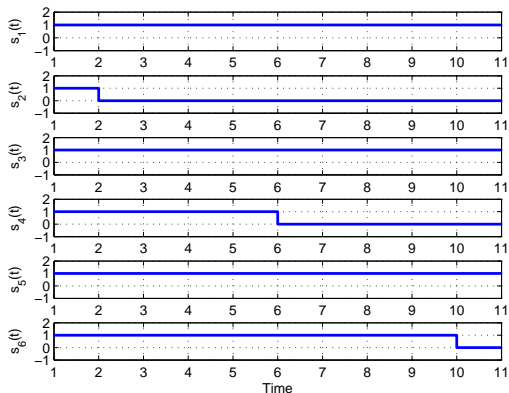


Figure: States vs. time for 6 clusters located equidistantly on a line

- The algorithm converges to the optimal configuration in this example.

Outline

- 1 Introduction
- 2 The GADIA Algorithm
- 3 Performance Analysis**
 - Static Case
 - Time-varying Case
 - Spectrum Sensing Time
 - Adaptive Rate
- 4 Conclusions

Convergence

Theorem

Given any reciprocal channel model, the GADIA Algorithm converges to a local minimum in polynomial time in N .

Outline of the Proof

- If a cluster c_i updates its frequency band, it reduces the amount of interference it experiences.
- Due to channel reciprocity, the amount of interference experienced by all the other clusters due to c_i is also reduced.
- Thus, the aggregate interference of the network is a non-increasing function of time.
- Since the aggregate interference is also lower-bounded, the algorithm converges to a local minimum.

Outline of the Proof (cont.)

- Assuming that the clusters are distributed in an area with diameter $O(Nd)$ for some constant d , the least amount of decrement in each update step is $O(\frac{1}{N\eta})$.
- The maximum of aggregate interference is $O(N)$.
- The algorithm converges in polynomial time in N .

Upper Bound on the Performance

Theorem

Let $I_a(N, \{d_{ij}\})$ denote the aggregate interference of all the clusters corresponding to the state of the algorithm following convergence and $I_w(N, \{d_{ij}\})$ to be the aggregate interference for the worst case interference scenario (all clusters transmitting in one frequency band), then

$$I_a(N, \{d_{ij}\}) \leq \frac{1}{r} I_w(N, \{d_{ij}\})$$

- Proof outline:

After convergence, c_i is in a frequency band, say $k \in \{1, 2, \dots, r\}$ such that $I_{c_i,k}(N, \{d_{ij}\}) \leq I_{c_i,j}(N, \{d_{ij}\})$, for all $j \neq k$. Therefore,

$$rI_a(N, \{d_{ij}\}) = r \sum_i I_{c_i,k}(N, \{d_{ij}\}) \leq \sum_i \sum_j I_{c_i,j}(N, \{d_{ij}\}) = I_w(N, \{d_{ij}\})$$

Lower Bound for the Optimal Strategy (For Linear Arrays)

Definition

A Linear Array is an array of clusters for which all the clusters are co-linear (lie on a line).

Theorem

For a Linear Array of clusters in $[0, (N - 1)d]$, we have

$$\lim_{N \rightarrow \infty} \frac{1}{N} I_o(N, \{d_{ij}\}) \geq \frac{1}{r^\eta} 2\zeta(\eta) \frac{\tilde{P}}{d^\eta}$$

where $I_o(N, \{d_{ij}\})$ is the aggregate interference of the optimal strategy, \tilde{P} is the transmission power of one node and $\zeta(\eta)$ is the Riemann zeta function.

Outline of the proof

- Motivation: The optimal frequency band assignment strategy is not known for general linear arrays. We therefore try to lower bound the aggregate interference of the optimal assignment.
- It can be shown that the minimum aggregate interference of any linear array of N clusters in $[0, (N - 1)d]$, is higher than that of the corresponding *uniform* linear array, where the N clusters are located in $[0, (N - 1)d]$ equidistantly, for N large enough.
- The righthand side corresponds to the minimum normalized aggregate interference of a uniform linear array.
- Uniform linear array achieves the bound, for N large enough.
- The inequality holds for any linear array.

The case of $r = 2$

Theorem

If $r = 2$ and $\eta \geq 2$, then the optimal strategy for a uniform linear array is the alternating assignment of the two frequency bands, for any N .

- Outline of the proof:
 - ▶ For $\eta \geq 2$, there can not be any 3 successive clusters in the same frequency band, in the optimal configuration.
 - ▶ There is a sequence of changes in the assignments for any given configuration, which results to the alternating assignment and in each step, the aggregate interference decreases.

Performance Result for Linear Arrays

Corollary

For a linear array in $[0, (N - 1)d]$, we have

$$\frac{I_a(N, \{d_{ij}\})}{I_o(N, \{d_{ij}\})} \leq \frac{r^{\eta-1}}{\left(\frac{d_{\min}}{\min\{d_{\max}, d\}}\right)^\eta}$$

as $N \rightarrow \infty$, where $I_a(N, \{d_{ij}\})$ denote the aggregate interference of all the clusters corresponding to the state of the algorithm following convergence and $I_o(N, \{d_{ij}\})$ is the aggregate interference corresponding to the optimal strategy.

- Proof outline:

- ▶ Combining the previous theorems.
- ▶ It is a worst-case bound. Applicable to any linear array.

Remarks

- The optimal strategy is non-trivial for a general uniform array of clusters.
- The alternating strategy for finite N and $r > 2$ seems to be the optimal strategy, although it is not trivial.
- The Riemann zeta function in the lower bound expression is merely a consequence of the path loss model and the fact that at each time slot, only one transmitter and one receiver are active in each cluster.

Simulation Setup

- Simulations for $r = 2$, $d = 1$ and $\eta = 2$.
- We let all the clusters to be in frequency band b_1 initially.
- We repeat the updates until the convergence is achieved. The simulations are obtained by ensemble averaging over 100 different update patterns.
- The i th cluster is distributed uniformly on the interval $[i - 0.5 + G, i + 0.5 - G]$, where G denotes the guard-band to restrict the minimum distance of two adjacent clusters.

Simulation Results: 1D Arrays

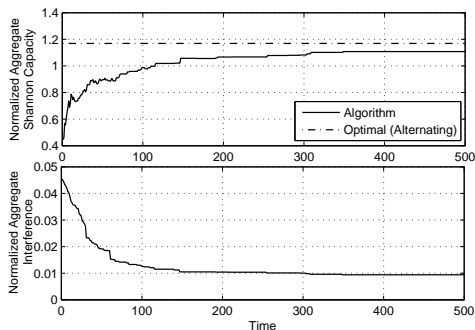


Figure: Average Normalized Aggregate Shannon Capacity and Normalized Aggregate Interference (dB) vs. time for a uniform linear array of 100 clusters

- The position of each cluster is distributed uniformly on the interval $[-0.4, 0.4]$ around the sites of the \mathbb{Z}_1 lattice (Monte Carlo sampling using 500 ensembles).
- The algorithm achieves about 94.8% of the Shannon Capacity corresponding to the optimal (alternating) frequency band assignment ($r = 2$).

Simulation Results: 1D Arrays

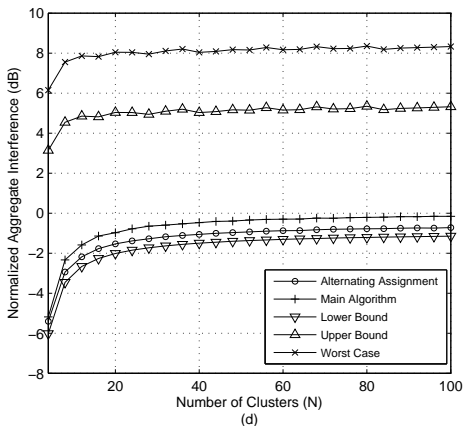


Figure: Normalized Aggregate Interference (dB) vs. N for a non-uniform linear array with $G = 0.02$

Here, the assumption on the size of the clusters being much smaller than the distances between them is relaxed ($G = 0.02$). The algorithm still performs within 1.5 dB of the optimal strategy

Simulation Results: 2D Arrays

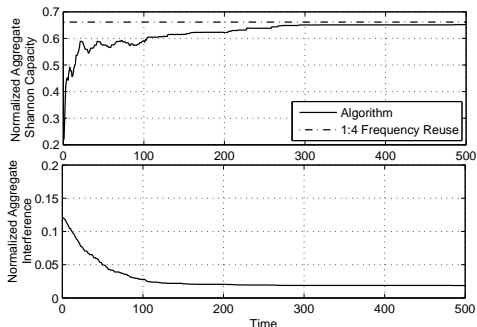
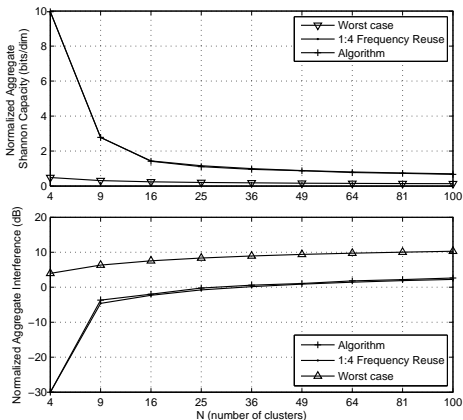


Figure: Average Normalized aggregate Shannon capacity and normalized aggregate interference curves for a rectangular lattice of 100 clusters vs. t

- The position of each cluster is distributed uniformly in a 0.8×0.8 rectangle around the sites of the \mathbb{Z}_2 lattice (Monte Carlo sampling using 500 ensembles).
- The algorithm achieves about 98.5% of of the Shannon Capacity corresponding to the near-optimal 1:4 frequency reuse ($r = 4$).

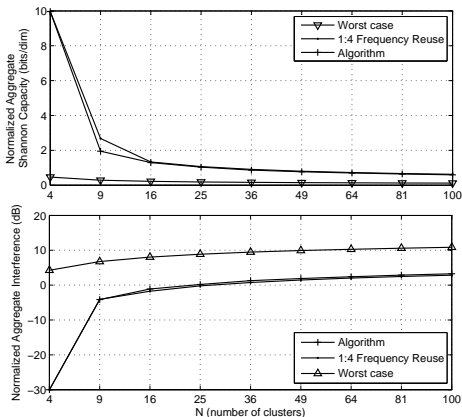
Simulation Results: Rectangular Lattice



(a)

Figure: Normalized aggregate Shannon capacity and normalized aggregate interference curves for a rectangular lattice vs. N

Simulation Results: Hexagonal Lattice



(b)

Figure: Normalized aggregate Shannon capacity and normalized aggregate interference curves for a hexagonal lattice vs. N

Time-varying setup

- The clusters go on and off over time, according to a two-state Markov model.
- For cluster c_i , $i = 1, 2, \dots, N$, we consider an activity indicator state $a_i(l)$, such that $a_i(l) = 1$ and $a_i(l) = 0$ correspond to being active and inactive at time l , respectively. Let $P_0^{c_i}(l)$ and $P_1^{c_i}(l)$ be the probability of c_i being in activity indicator state 0 and 1 at time l , respectively. The evolution of the probabilities is given by:

$$\begin{pmatrix} P_0^{c_i}(l+1) \\ P_1^{c_i}(l+1) \end{pmatrix} = \begin{pmatrix} \alpha & 1 - \alpha \\ 1 - \alpha & \alpha \end{pmatrix} \begin{pmatrix} P_0^{c_i}(l) \\ P_1^{c_i}(l) \end{pmatrix}$$

Assumptions

- The algorithm converges to an equilibrium point on average, in a statistical sense.
- We assume that the clusters update their frequency band asynchronously according to the same temporal statistics.
- In a continuous approximation, let $I(t)$ denote the aggregate interference of the network at time t .
- The update process is modeled by a Poisson process of rate $\frac{1}{\Delta T}$, *i.e.*, each cluster updates its frequency band with a rate $\frac{1}{N\Delta T}$.
- For the moment, we assume that all the N clusters are active.
- We associate $\epsilon_i = -1$ and $\epsilon_i = 1$ to clusters in band b_0 and b_1 , respectively.
- All the following analysis is valid for steady state (near equilibrium) and assuming that the number of users switching on/off in each time slot is much less than the total number of users.

Dynamics of the Algorithm for $r = 2$

- An active cluster experiences an interference of $\frac{1}{2}(I_i - \sum_{j \neq i} \frac{\epsilon_j(t)}{d_{ij}^\eta})$ or $\frac{1}{2}(I_i + \sum_{j \neq i} \frac{\epsilon_j(t)}{d_{ij}^\eta})$ depending on the band it is using, where I_i is the worst case interference experienced by cluster c_i .
- We define the band b_j to be *appropriate* for cluster c_i , if c_i is assigned in band b_j in the optimal strategy.
- If a total of M users $c_i, i \in \{i_1, \dots, i_M\}$ are not in the *appropriate* frequency bands, the aggregate interference will be

$$I(t) = I_a + \sum_{k=1}^{M(t)} \left| \sum_{j \neq i_k} \frac{\epsilon_j(t)}{d_{i_k, j}^\eta} \right|$$

where I_a is the target performance of the algorithm.

- Assuming Ergodicity, we have

$$\mathcal{E}[I(t)] - \mathcal{E}[I_a] = 2\mathcal{E}[M(t)]\mathcal{E}\left[\left| \sum_{j \neq i} \frac{\epsilon_j(t)}{d_{ij}^\eta} \right|\right]$$

where \mathcal{E} denotes the ensemble average.

Dynamics of the Algorithm

- For any update, the average change in $\mathcal{E}[I(t)]$, will be

$$\Delta\mathcal{E}[I(t)] = \frac{\rho\mathcal{E}[M(t)]}{N} 2\mathcal{E}\left[\left|\sum_{j\neq i} \frac{\epsilon_j(t)}{d_{ij}^\eta}\right|\right]$$

where ρ is a geometrical constant showing the effective number of interacting neighbors to a cluster including itself (this is a linearization near the equilibrium point).

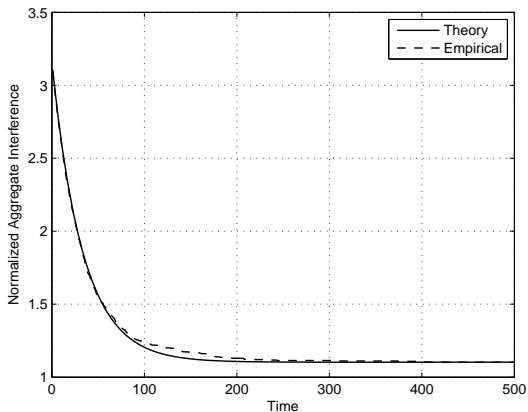
- Combining the above equations we get

$$\frac{\Delta\mathcal{E}[I(t)]}{\Delta T} = -\frac{\rho}{N\Delta T}(\mathcal{E}[I(t)] - \mathcal{E}[I_a])$$

- Spatial ergodicity is assumed in the derivation of the above dynamics.
- On the time scale of the updates, using the ansatz $N\Delta T \triangleq \tau$ one can write

$$\frac{d\mathcal{E}[I(t)]}{dt} = -\frac{\rho}{\tau}(\mathcal{E}[I(t)] - \mathcal{E}[I_a])$$

Simulation Results: Dynamics of the Algorithm for a Uniform Linear Array of 100 clusters



- A theoretical estimate for ρ in this case is 3, since every cluster has two nearest neighbors.

Time-varying Statistics

- We can model the change in the number of active clusters by two Poisson counters of rate λ .
- Each cluster when activated, approximately experiences the instantaneous normalized aggregate interference of the network (Ergodicity).
- If we define $\mathcal{I}(t) \triangleq \mathcal{E}[I(t)]$ and $\mathcal{I}_a(t) \triangleq \mathcal{E}[I_a(t)]$, under the assumption of λ being small compared to $\frac{1}{\tau}$, we have the new dynamics in the Itô form as

$$d\mathcal{I}(t) = -\frac{\rho}{\tau}(\mathcal{I}(t) - \mathcal{I}_a(t))dt + \frac{4}{N}\mathcal{I}(t)(dN_+ - dN_-)$$

where dN_+ and dN_- are two independent Poisson counters of rate λ , *i.e.*, $E(dN_+) = E(dN_-) = \lambda dt$.

- Under this model, the algorithm converges in mean to the target performance.

Steady State Analysis

- The variance equation associated with the dynamics is

$$\frac{dE(\mathcal{I}^2(t))}{dt} = -\left(\frac{2\rho}{\tau} - \frac{32\lambda}{N^2}\right)E(\mathcal{I}^2(t)) + \frac{2\rho}{\tau}\mathcal{I}_a^2$$

- In the steady state, the variance settles down to

$$\sigma_{ss}^2 = \mathcal{I}_a^2 \frac{16\lambda\tau}{N^2\rho - 16\lambda\tau}$$

given $\frac{16\lambda\tau}{N^2\rho} < 1$.

- According to our model for cluster activities, $\lambda = N^2(1 - \alpha)/2\tau$. Thus, we get the following trade-off inequality

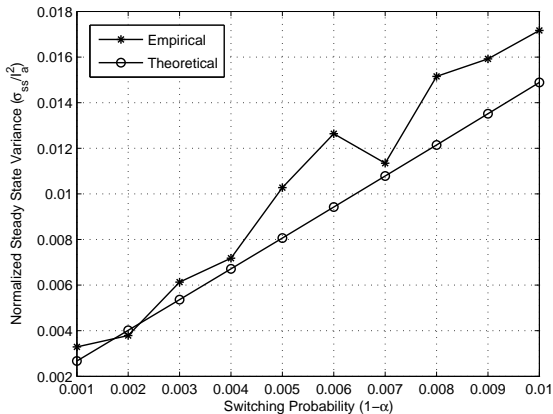
$$\frac{8(1 - \alpha)}{\rho} < 1$$

in order to have a finite variance in the steady state.

Remarks

- The model gives a simple trade-off inequality for design purposes.
- The geometrical parameter ρ can be empirically estimated for different network topologies. However, theoretical estimates are possible.
- If $\lambda = O(N^{1-\epsilon})$ for some $\epsilon > 0$, as $N \rightarrow \infty$, the inequality always holds. Therefore, the algorithm converges in both mean and variance in the sub-linear regime.
- The analysis can be generalized to other statistical models for the activity of the clusters over time.

Simulation Results: Empirical vs. Theoretical Steady State Variance



- The model matches the empirical data (averaged over 500 ensembles).

Simulation Results: Capacity vs. Time-varying Activities

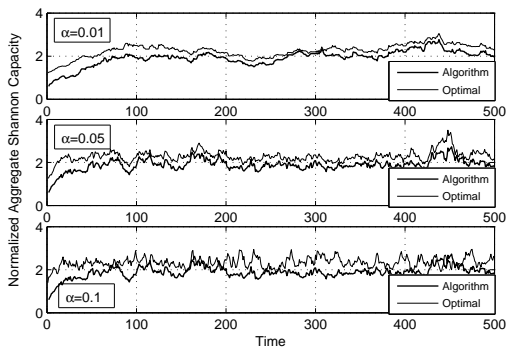


Figure: Normalized Aggregate Shannon Capacity for a linear array of 100 clusters vs. time, for on/off switching probabilities $\alpha = 0.01, 0.05$ and 0.1 .

- For on/off switching probabilities $\alpha = 0.01, 0.05$ and 0.1 ($N\alpha = 1, 5$ and 10) the algorithm achieves about 90%, 86% and 83% of the optimal capacity on average, respectively.

Impact of Spectrum Sensing Time

- In the previous analysis, we assumed that at each instance of time only one cluster is sensing the spectrum in order to update its frequency band.
- Suppose that each cluster requires a time interval of $T_0 \ll \Delta T$ to perform the spectrum sensing, where $1/N\Delta T$ is the update rate of each cluster.
- A cluster can not sense the interference of other clusters who are simultaneously sensing the spectrum, which may result in a wrong update decision.

Probability of Simultaneous Sensing

- Suppose that at time t , cluster c_i is sensing the spectrum to update its frequency band. The probability that in the interval $[t, t + T_0]$ another cluster is also sensing the spectrum is given by

$$\begin{aligned}\beta &\triangleq \binom{N-1}{1} \left(\lambda_0 T_0 e^{-\lambda_0 T_0} \right) \left(e^{-\lambda_0 T_0} \right)^{N-2} \\ &= (N-1) \lambda_0 T_0 e^{-(N-1) \lambda_0 T_0} = \frac{N-1}{N} \frac{T_0}{\Delta T} e^{-\left(\frac{N-1}{N}\right) \frac{T_0}{\Delta T}} \\ &\approx \frac{T_0}{\Delta T}\end{aligned}$$

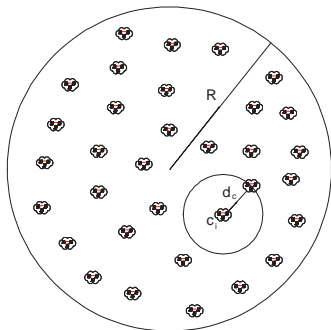
where the approximation is based on the assumptions $T_0 \ll \Delta T$ and $N \gg 1$.

Simultaneous Sensing (cont.)

- Interference gap between the two bands b_0 and b_1 experienced by cluster c_i is given by

$$\frac{P_0}{d_c^n(t)} \triangleq \mathcal{E} \left[\left| \sum_{j \neq i} \frac{P_0 \epsilon_j(t)}{d_{ij}^n} \right| \right]$$

- If a cluster at a distance at most $d_c(t)$ from c_i is sensing the spectrum, and is in the frequency band corresponding to the updated frequency band of c_i , then c_i may make a wrong decision.



Probability of Making a Wrong Update

- Assuming that the clusters are distributed homogeneously in two dimensions, we can write the probability of a wrong decision as follows:

$$p_e(t) \triangleq \Pr(\text{wrong decision by } c_i \text{ at time } t) = \frac{1}{2} \beta \frac{\pi d_c^2(t)}{\pi R^2}$$

- R corresponds to the area over which the clusters are distributed.
- The factor of $1/2$ in the above equation reflects the fact that only the clusters in the inappropriate band of c_i may cause a wrong decision.

Convergence

Lemma

The algorithm converges to an equilibrium point with probability 1, given

$$\beta \leq \frac{2I_0}{I_0 + I_m} \quad (1)$$

where

$$I_0 \triangleq \frac{P_0}{d_{max}^\eta} \quad (2)$$

$$d_{max} \triangleq \max_{i,j} d_{ij} \quad (3)$$

and

$$I_m \triangleq \max_i \sum_{j \neq i} \frac{P_0}{d_{ij}^\eta} \quad (4)$$

Modeling the Dynamics

- The spectrum sensing collisions will result in fluctuations around the equilibrium point.
- In order to model the behavior of the algorithm near the equilibrium point, we define two Poisson counters dN_1 and dN_2 , with

$$E[dN_1] = \frac{1 - p_e(t)}{\Delta T} dt$$

$$E[dN_2] = \frac{p_e(t)}{\Delta T} dt$$

- In other words, dN_1 corresponds to successful updates which has rate $(1 - p_e(t))/\Delta T$ and dN_2 corresponds to wrong updates which has rate $p_e(t)/\Delta T$.

Modeling The Dynamics (cont.)

- One can write the changes in the aggregate interference using these counters as follows:

$$d\mathcal{E}[I] = -\frac{\rho\mathcal{E}[M(t)]}{N}2\Delta I(t)dN_1 + \left(1 - \frac{\rho\mathcal{E}[M(t)]}{N}\right)2\Delta I'(t)dN_2$$

where $\Delta I(t) \triangleq \mathcal{E}\left[\left|\sum_{j \neq i} \frac{P_0 \epsilon_j(t)}{d_{ij}^\alpha}\right|\right]$ corresponds to the average decrement due to a successful update by a cluster whose switching bands decreases the aggregate interference, $\Delta I'(t)$ corresponds to the average increment due to a wrong update by a cluster whose switching bands increases the aggregate interference.

- The above equation can be simplified as

$$d\mathcal{E}[I(t)] = -\frac{\rho}{N}(\mathcal{E}[I(t)] - \mathcal{E}[I_a])dN_1 + \left(1 - \frac{\rho\mathcal{E}[M(t)]}{N}\right)2\Delta I'(t)dN_2$$

Steady State Variance

- By defining $f(t) \triangleq \left(1 - \frac{\rho \mathcal{E}[M(t)]}{N}\right) \Delta I'(t)$, the variance equation corresponding to the dynamics becomes

$$\begin{aligned} d\mathcal{I}^2(t) &= \left[\left(\mathcal{I}(t) - \frac{\rho}{N} (\mathcal{I}(t) - \mathcal{I}_a) \right)^2 - \mathcal{I}^2(t) \right] dN_1 \\ &+ \left[(\mathcal{I}(t) + 2f(t))^2 - \mathcal{I}^2(t) \right] dN_2 \end{aligned}$$

where $\mathcal{I}(t) \triangleq \mathcal{E}[I(t)]$ and $\mathcal{I}_a \triangleq \mathcal{E}[I_a]$

- Taking expectations and computing the steady state variance yields

$$\begin{aligned} \frac{dE[\mathcal{I}^2(t)]}{dt} &= - \left(\frac{2\rho}{N\Delta T} - \frac{\rho^2}{N^2\Delta T} \right) (1 - p_e(t)) E[\mathcal{I}^2(t)] \\ &+ \mathcal{I}_a \left(\frac{2\rho}{N\Delta T} E[\mathcal{I}(t)] + \frac{\rho^2}{N^2\Delta T} (\mathcal{I}_a - 2E[\mathcal{I}(t)]) \right) (1 - p_e(t)) \\ &+ \left(4f(t)^2 + 4E[\mathcal{I}(t)]f(t) \right) \frac{p_e(t)}{\Delta T} \end{aligned}$$

Steady State Variance (cont.)

- The steady state variance of the aggregate interference can be written as

$$\sigma_{ss}^2 \triangleq \lim_{t \rightarrow \infty} E \left[\left(\mathcal{I}(t) - E[\mathcal{I}(t)] \right)^2 \right] = \lim_{t \rightarrow \infty} \frac{2f(t)^2 p_e(t)}{\rho/N}$$

up to the first order in $p_e(t) \ll 1$ and $\rho/N \ll 1$.

- By simplifying and using proper upper bounds on $f(t)$, we get

$$\sigma_{ss}^2 < \beta \left(\frac{I_m^2}{\rho/N} \right) \left(\frac{NP_0}{I_w - \mathcal{I}_a} \right)^{2/\eta} \frac{1}{R^2}$$

where $I_m \triangleq \max_i \sum_{j \neq i} \frac{P_0}{d_{ij}^\eta}$ and $I_w \triangleq \sum_i \sum_{j \neq i} \frac{P_0}{d_{ij}^\eta}$.

Simulation Results: Steady State Variance

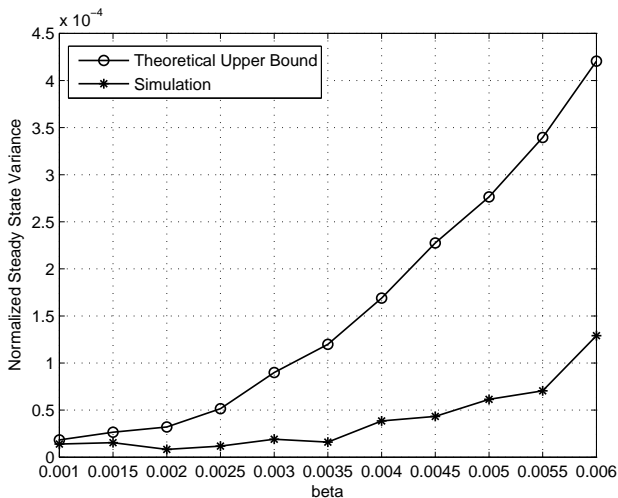


Figure: Normalized steady state variance of the equilibrium point for a rectangular array of 100 clusters vs. β

Adaptive Update Rates

- In previous sections, all the clusters update their frequency band with a constant rate.
- In order to do the updates in a more opportunistic way, each cluster c_i tosses a biased coin with head probability

$$p_i \triangleq \frac{|I_{c_i,0} - I_{c_i,1}|}{\max\{I_{c_i,0}, I_{c_i,1}\}} \quad (5)$$

where $I_{c_i,0}$ and $I_{c_i,1}$ denote the interference experienced in bands b_0 and b_1 , respectively.

- Each cluster updates its frequency band, if the outcome of the coin toss is head.
- In this manner, the less a cluster benefits from updating, the less frequent it updates its band.

Simulation Results: 1D Array, 2 Frequency Bands

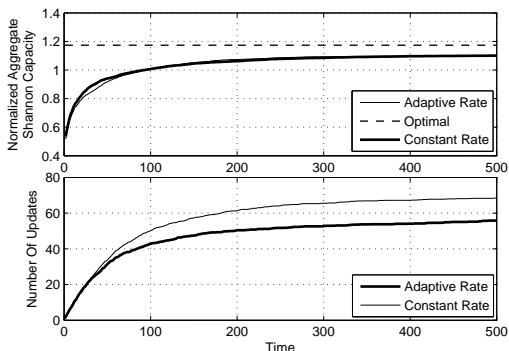


Figure: Average normalized aggregate Shannon capacity and number of updates for linear arrays of 100 clusters vs. time

- All the clusters are initially in band b_0 . Both schemes have similar performance, but the adaptive scheme reduces the number of updates by 20% (both achieve around 95% of the optimal aggregate Shannon capacity).

Outline

- 1 Introduction
- 2 The GADIA Algorithm
- 3 Performance Analysis
 - Static Case
 - Time-varying Case
 - Spectrum Sensing Time
 - Adaptive Rate

- 4 Conclusions

Conclusion

- Proposed a distributed algorithm for finding a sub-optimal frequency band allocation to the clusters in a network.
- Proved the convergence of the algorithm for any reciprocal channel model.
- Obtained performance bounds for one dimensional linear arrays of clusters (the algorithm outperforms the bounds).
- Evaluated the performance of the algorithm when clusters can be in sleep or active mode and go off and on according to time-varying statistics.
- Evaluated the performance of the algorithm under the impact of simultaneous spectrum sensing by different clusters.
- Considered adaptive update rates.

Future Work

- Finding performance bounds for more general network topologies.
- Generalization of the performance bounds to higher dimensions.
- Distributed control methods using limited feedback.
- Extension to vertical spectrum sharing.
- Allowing inter-cluster communications.
- Optimal adaptive update rates.