Wireless Pers Commun (2011) 57:5–18
DOI 10.1007/s11277-010-0003-y

Evaluation of Dynamic Channel and Power Assignment for Cognitive Networks

Juan D. Deaton · Syed A. Ahmad · Umesh Shukla · Ryan E. Irwin · Luiz A. DaSilva · Allen B. MacKenzie

Published online: 26 May 2010

Abstract In this paper, we develop a unifying optimization formulation to describe the Dynamic Channel and Power Assignment (DCPA) problem and an evaluation method for comparing DCPA algorithms. DCPA refers to the allocation of transmit power and frequency channels to links in a cognitive network so as to maximize the total number of feasible links while minimizing the aggregate transmit power. We apply our evaluation method to five

This work is partially supported by the Idaho National Laboratory (INL) Ph.D. Candidate Program and Virginia Tech Bradley Fellowship. Work supported by the INL is done under Battelle Energy Alliance, LLC under Contract No. DE-AC07-05ID14517 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a nonexclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Department of Energy or the U.S. Government.

J. D. Deaton (✉) · S. A. Ahmad · U. Shukla · R. E. Irwin · L. A. DaSilva · A. B. MacKenzie
Virginia Polytechnic Institute & State University, Blacksburg, VA, USA
e-mail: deatjd@vt.edu

S. A. Ahmad
e-mail: saahmad@vt.edu

U. Shukla
e-mail: ushukla@vt.edu

R. E. Irwin
e-mail: reirwin@vt.edu

A. B. MacKenzie
e-mail: mackenab@vt.edu

J. D. Deaton
Idaho National Laboratory, Idaho Falls, ID, USA

L. A. DaSilva
CTVR, Trinity College Dublin, Dublin, Ireland
e-mail: ldasilva@vt.edu

representative DPCA algorithms proposed in the literature. This comparison illustrates the tradeoffs between control modes (centralized versus distributed) and channel/power assignment techniques. We estimate the complexity of each algorithm. Through simulations, we evaluate the effectiveness of the algorithms in achieving feasible link allocations in the network, and their power efficiency. Our results indicate that, when few channels are available, the effectiveness of all algorithms is comparable and thus the one with smallest complexity should be selected. The Least Interfering Channel and Iterative Power Assignment algorithm does not require cross-link gain information, has the overall lowest run time, and achieves the highest feasibility ratio of all the distributed algorithms; however, this comes at a cost of higher average power per link.

Keywords Cognitive networks · Dynamic channel and power assignment · Mobile adhoc networks · Network density

1 Introduction

In a Cognitive Network (CN), radios adapt their operating parameters to achieve network-wide objectives such as connectivity and efficient resource utilization [7]. In Dynamic Spectrum Access (DSA), a CN of spectrum agile radios must efficiently utilize spectrum resources throughout the network. Research on Dynamic Channel and Power Assignment (DCPA) seeks effective ways in which a CN of autonomous radios can assign an appropriate frequency channel and transmit power to improve connectivity and spectral efficiency given available spectrum. Many DCPA techniques have been proposed in the literature [1–5,8], and evaluating the tradeoffs among these techniques from the existing body of work is a difficult task. Each DCPA algorithm can be evaluated under different topologies, node densities, and metrics. The contribution of this paper is to propose a unifying framework with which to evaluate the performance of DCPA mechanisms and to apply this framework in the comparison of five DCPA algorithms that are representative of the existing literature.

In the next section, we provide a mathematical formulation of DCPA as an optimization problem and introduce notation adopted throughout the paper. Then, in Sect. 3, we summarize each of the five algorithms, based on [1,2,4,5], and an additional algorithm proposed by us. In Sect. 3 we also present a complexity analysis for each algorithm. Section 4 describes the method and metrics for the performance evaluation of the five DCPA algorithms. Section 5 presents the results of this evaluation: a comparative performance study under a common set of assumptions and conditions. Finally, we discuss our main conclusions in Sect. 6, where we also provide thoughts on open issues for channel assignment.

2 System Model and Problem Formulation

We consider a CN of spectrum agile radios that seek to create a self-organized topology through frequency channel selection and power control. In our system model, we define a set $\mathcal{C}$ of frequency channels. Additionally, we define a set $\mathcal{L}$ of communications links, where a link comprises a transmitter and receiver. Links seek to communicate by using a frequency channel $c \in \mathcal{C}$.

Given a set of communications links operating on a channel c, $\mathcal{L}_c$, the Signal to Interference and Noise Ratio (SINR) of the receiver of link $i \in \mathcal{L}_c$, γ_i, is determined by:

$$\gamma_i = \frac{G_{i,i} P_i}{N_o + \sum_{j \in \mathcal{L}_c, j \neq i} G_{j,i} P_j}. \tag{1}$$

$G_{j,i}$ is the gain between the transmitter of link j and the receiver of link i. The variable P_i denotes the power of the transmitting node of link i and N_o the thermal noise. The receiver of each link requires a minimum SINR β for the link to be feasible.

Given a set of transmitter–receiver pairs and corresponding feasibility constraints, the DCPA problem is to allocate limited resources (i.e., channels and transmit power) to these pairs to maximize the total number of feasible links and, for these links, to minimize the aggregate transmit power. Using this definition, we formulate a unifying optimization problem to describe the objective of the DCPA algorithms studied in this paper:

$$\text{Maximize}: \quad M \sum_{i \in \mathcal{L}} \sum_{c \in \mathcal{C}} l_i^c - \sum_{i \in \mathcal{L}} P_i, \tag{2}$$

$$\text{Subject to}: \quad \sum_{c \in \mathcal{C}} l_i^c \leq 1 \qquad \forall i \in \mathcal{L}, \tag{3}$$

$$P_i \geq l_i^c \beta \left(\frac{N_o}{G_{i,i}} + \sum_{j \in \mathcal{L}, j \neq i} \frac{G_{j,i}}{G_{i,i}} P_j l_j^c \right) \quad \forall i \in \mathcal{L}, c \in \mathcal{C}, \tag{4}$$

$$0 \leq P_i \leq P_{\max} \quad \forall i \in \mathcal{L}. \tag{5}$$

The optimization variable l_i^c reflects the assignment of channel c to link i as described by:

$$l_i^c = \begin{cases} 1 & \text{if link } i \text{ is assigned channel } c \in \mathcal{C} \\ 0 & \text{otherwise.} \end{cases}$$

M is a weighting factor that, when sufficiently large, prioritizes maximizing the number of feasible links over minimizing the total transmit power in the network. The constraint expressed in inequality (3) restricts a link to only one channel. Since we are minimizing transmit power, inequality (4) allows the transmit power to be set to zero if the link is not assigned a channel. Otherwise, the link is required to meet the minimum SINR requirement β. Inequality (5) constrains the maximum transmitter power to $P_{\max}$.

3 Algorithms

After a review of the DCPA literature on multi-channel ad hoc networks, we adapted algorithms based on the works of [1,2,4,5] for our comparative analysis. We adapted some of the underlying assumptions of each work to allow for equitable comparison, while maintaining their unique algorithmic features. In light of changes from their original work in [1,2,4,5], these adapted algorithms are renamed as: Least Interfering Channel and Non-Iterative Power Assignment (LICNPA), Spatial Channel Separation and Iterative Power Assignment (SCSI-PA), Least Interfering Channel and Iterative Power Assignment (LICIPA), Minimum Power Increase Assignment (MPIA), and Conflict Graph Assignment (CGA). We also propose a new distributed algorithm LICIPA, which combines mechanisms from [2,4]. We selected these algorithms to illustrate the tradeoffs between different control modes (centralized versus distributed) and among assignment techniques as presented in each algorithm description. Additionally, we develop a complexity analysis for each of the five algorithms.

3.1 Least Interfering Channel and Non-Iterative Power Assignment (LICNPA)

In LICNPA, link i is assigned the channel that has the lowest measured interference at the receiver and below a threshold parameter, I_{th}. If no channel is below I_{th}, the link is infeasible. If a link is assigned a channel, the transmitter begins with initial power P_{ref}. If the SINR of the receiver is below β, the transmitter increases power in a one-step increment using the following equation:

$$P_i = \min\left(P_{\max}, P_{\text{ref}}\sqrt{\frac{I_{\text{th}}}{I_i}\frac{\beta}{\gamma_i}}\right), \tag{6}$$

where I_i is the measured interference power at the receiver of link i. If the SINR of the receiver of link i is still less than β after the power increase by Eq. 6, the link is considered infeasible. According to [4], the objective of Eq. 6 is to prevent subsequent admitted links from increasing interference such that the SINR of active co-channel links will drop below β. Additionally, this power control scheme never reduces transmit power, making the overall system performance sensitive to P_{ref}.

In terms of complexity for LICNPA, links are admitted sequentially using a two-step operation for every link, $O(L)$. First, LICNPA must determine the Least Interfering Channel (LIC) for channel assignment, $O(C)$, and adjust power of the link, $O(1)$. This results in a complexity of $O(CL)$ for LICNPA. In summary, using the LIC requires less power for the SINR to remain above β, thus minimizing the transmit power of each link and maximizing the number of feasible links.

3.2 Spatial Channel Separation and Iterative Power Assignment (SCSIPA)

Unlike the original proposal of [2], SCSIPA assumes frequency channels and distributed power control, as opposed to time slots and centralized power control. SCSIPA also assumes a common control channel in which nodes exchange location and channel assignment information with one another. Using the location and channel assignment information, the transmitter node of link i determines which receivers are within distance $d_{i,i}$, where $d_{i,i}$ is the Euclidean distance between the transmitter and receiver of link i. The transmitter of link i then randomly selects a channel c not being used by the neighboring receivers within distance $d_{i,i}$.

After the channel assignment for link i, the transmitter begins transmitting with initial power parameter P_{ref}. All transmitters on channel c then iteratively adjust their transmit power according to:

$$P_i\,(k+1) = \min\left(P_{\max}, \frac{\beta}{\gamma_i}P_i\,(k)\right), \tag{7}$$

where k is the iteration number. Foschini in [3] demonstrated that when transmitters use Eq. 7 to adjust their power levels, the transmit powers of the links will converge exponentially. In SCSIPA, if the link cannot maintain an SINR of at least β without the transmit power of the link exceeding $P_{\max}$, the link is infeasible and the transmit power of the link is set to zero. This procedure is performed sequentially for each link. This power assignment technique is unlike the centralized power control of [2], in which transmitter power levels are coordinated and assigned simultaneously. In summary, SCSIPA manages interference among co-channel links by spatially separating interferers and using power control in Eq. 7 to minimize transmitter power.

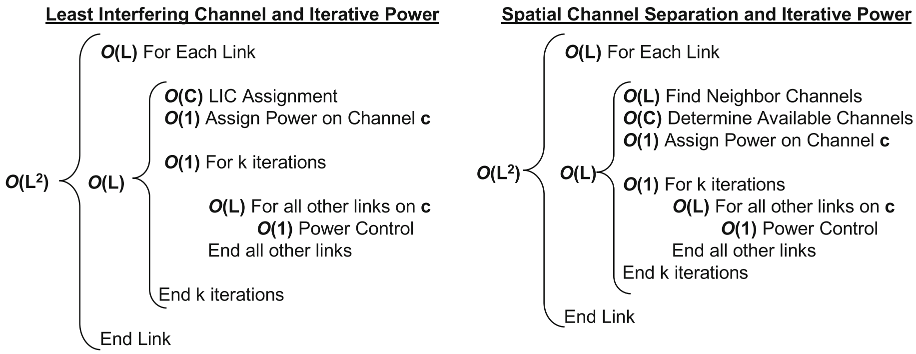


Fig. 1 Complexity derivation for LICIPA and SCSIPA

Our derivation of the complexity of the SCSIPA is shown in Fig. 1. SCSIPA performs channel assignment and power control for each link, $O(L)$. The link must discover the channel assignment of the $d_{i,i}$ neighbors by control messages from at most L links, $O(L)$. The link then determines and selects a channel not used by the $d_{i,i}$ neighbors, $O(C)$. After the initial power is assigned to a link, transmit power levels are allowed to converge for, at most, L links, $O(L)$. Since we assume $L \gg C$, the dominant operation is determining neighbor channel information and allowing power levels to converge for both $O(L)$. Since this procedure is performed for every link, the complexity is $O(L^2)$ for this decentralized algorithm.

3.3 Least Interfering Channel and Iterative Power Assignment (LICIPA)

LICIPA is an algorithm created by combining power control from SCSIPA and channel assignment from LICNPA. In LICIPA, a link is assigned to the LIC as long as the received power of the LIC is below I_{th}. If no channel has received interference power below I_{th}, the link is infeasible. If a link is assigned a channel, the transmitter begins with initial power P_{ref}. After channel assignment, Eq. 7 is used by the links to iteratively adjust transmit power. If a link cannot maintain a SINR level of β without the transmit power of the link exceeding $P_{\max}$, the link is infeasible and the transmit power of the link is set to zero. In this algorithm, each link is admitted sequentially. LICIPA uses the LIC and the transmit power of Eq. 7 for maximizing the number of feasible links and minimizing transmit power.

Figure 1 shows the complexity derivation for LICIPA. The complexity for this algorithm is similarly derived as for SCSIPA and LICNPA. Determining and assigning the LIC is $O(C)$. Power control is then iterated for all other links on the channel, $O(L)$. Since we assume that $C < L$, $O(L)$ dominates in the inner loop and the algorithm has a complexity of $O(L^2)$.

3.4 Minimum Power Increase Assignment (MPIA)

MPIA is a centralized algorithm based on the *Minimum Incremental Power Algorithm* from [5]. In MPIA, global knowledge of cross-link gains is used to determine channel and power assignments. Using the cross-link gains, MPIA determines the feasibility of adding a new link into set $\mathcal{L}_c$, and if the new link is feasible, calculates the change in aggregate power ΔP_c that results from adding the new link into set $\mathcal{L}_c$ [8]. The feasibility test and calculation of ΔP_c are performed for all sets $\mathcal{L}_c$. MPIA then assigns the new link to the set $\mathcal{L}_c$ that yields the minimum ΔP_c. If the link is not assigned to any $\mathcal{L}_c$, then its transmit power is set to zero, and the link is declared infeasible. Unlike the original presentation of [5], MPIA requires

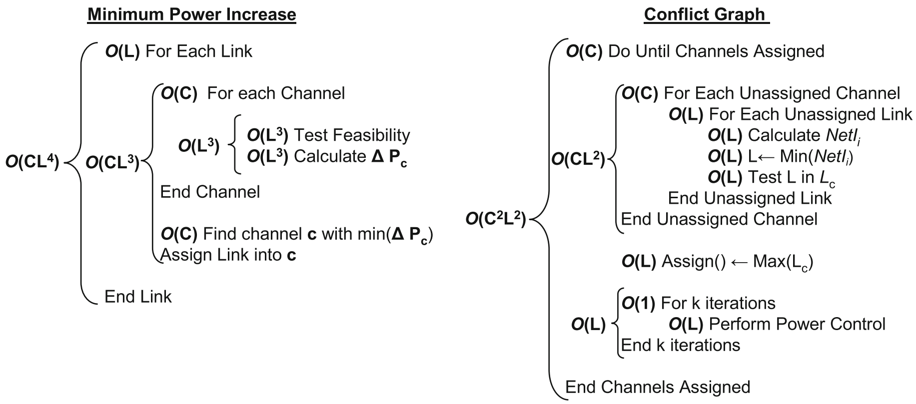


Fig. 2 Complexity derivation for MPIA and CGA

all links to meet the same minimum SINR requirement, β. Additionally, the order of link admittance is selected randomly. In summary, MPIA is a greedy assignment algorithm that seeks to maximize feasible link assignments using the minimum power increase on $\mathcal{L}_c$.

Our derivation of the complexity of MPIA is shown in Fig. 2. Given $|\mathcal{L}| = L$ links and $|\mathcal{C}| = C$ communications channels, MPIA must test every link $O(L)$, in every channel, $O(C)$. Testing the feasibility of adding a new link to $\mathcal{L}_c$ requires calculating the eigenvalue of a matrix whose dimensions are, at most, $L \times L$. The dominant operation in the eigenvalue calculation is the determinant, which has complexity of $O(L^3)$ [6]. ΔP_c requires calculating the inverse of a matrix whose dimensions are at most $L \times L$. By Gauss-Jordan elimination, the calculation of the matrix inverse has a complexity of $O(L^3)$ [6]. Both calculations for feasibility and ΔP_c are estimated for the worst case as $O(L^3)$. $O(L^3)$ dominates in the algorithm complexity, and combining the outer loops results in an algorithm complexity of $O(CL^4)$ for this centralized algorithm.

3.5 Conflict Graph Assignment (CGA)

CGA, based on [1], maximizes the number of feasible links through a greedy assignment algorithm using global knowledge of the cross-link gains in a weighted conflict graph. The algorithm begins by calculating the number of possible feasible links for each unassigned channel by attempting to place all links on each unassigned channel. After this calculation, CGA assigns the channel that supports the maximum number of feasible links. Power control is done subsequently to minimize total power consumption.

To calculate the number of feasible links for each unassigned channel, an adjacency matrix of weighted edges of the conflict graph is represented by:

$$\mathbf{G}(i, j) = \begin{cases} G_{i,j} & \text{if} \quad i \neq j \\ 0 & \text{if} \quad i = j. \end{cases} \tag{8}$$

Using $\mathbf{G}$, CGA calculates the potential network interference introduced by the transmitter of link i as $\text{Net}I_i = \sum_j G_{i,j}$. The link that has the min $(\text{Net}I_i)$ is tested for feasibility in $\mathcal{L}_c$. To test feasibility, each γ_i is then calculated with $P_i = P_{\max}$ in $i \in \mathcal{L}_c$. If the addition of the new link into $\mathcal{L}_c$ would not cause γ_i to drop below β for any $i \in \mathcal{L}_c$, the link can be added into set $\mathcal{L}_c$; otherwise, the link is not added. In either case, the link is discarded from $\mathbf{G}$, and $\mathbf{G}$ is recalculated to determine the next link with the min $(\text{Net}I_i)$. The process is repeated until all links are attempted. The number of feasible links for each unassigned channel is calculated in the same manner. The channel that supports the maximum number of links, max $|\mathcal{L}_c|$, is then assigned $\mathcal{L}_c$. This routine continues until all channels are assigned. If it is not possible to add a link into any channel, the link is infeasible.

While [1] does not use power control to minimize aggregate transmit power or reduce interference, it does suggest using the distributed power control mechanism in [3]. In CGA, upon completion of channel assignment, Eq. 7 is used to calculate the power of all links with initial power $P_i = P_{\max}$. Convergent power levels are then calculated for each link and assigned. In summary, CGA is a greedy assignment algorithm that uses the objective function in (2) by finding the maximum number of feasible links to each channel and minimizes the total transmit power by Eq. 7.

Our derivation of the complexity of CGA is shown in Fig. 2. CGA assigns links to a channel in every iteration, $O(C)$. For every unassigned channel, $O(C)$, CGA, in the worst case, tests every link, $O(L)$. For all links, the network interference, the minimum interferer, and the test for feasibility in $\mathcal{L}_c$ are calculated, each with complexity $O(L)$. Once the assignment for a particular channel is completed, $O(L)$, the power levels are calculated for all links in the channel, $O(L)$. The channel assignment operations executed for every link have a dominant complexity of $O(CL^2)$. Since this process is repeated until all channels are assigned, the worst-case complexity of CGA is $O(C^2L^2)$.

4 Evaluation Method and Metrics

The main goal of this work is to develop an evaluation method that can equitably compare distinct DCPA algorithms. To perform this comparison, the algorithms are given a set of L potential links in which they seek to fulfill the objective function defined in Eq. 2. We consider links that have a maximum separation distance between the transmitter and receiver of $d_{\max}$. Performance metrics are evaluated by varying the density of links, $d_{\max}$ and then number of channels.

We evaluate algorithm performance by varying link density because we seek to understand how well each algorithm is able to manage interference through channel assignment and power control. By increasing link density, we increase the aggregate interference experienced by each link. When density is low, the mean distance among potential interferers is larger than the intended transmitter-receiver distance. As the density increases, the distance between a transmitter and its intended receiver will approach the mean distance among potential interferers. Therefore, increasing the density provides a means to increase potential aggregate interference for each link.

Additionally, the choice of $d_{\max}$ plays a role in SINR being dominated by either noise or interference. If $d_{\max}$ is small, the SINR of each link will be dominated by noise. Conversely, if $d_{\max}$ is large, the SINR of each link will be dominated by interference. Additionally, if $d_{\max}$ is sufficiently large and the density sufficiently low, links could be infeasible because of attenuation from path loss. Therefore, two different values of $d_{\max}$ are used for this evaluation while varying the link density to explore these regions of interest.

For our algorithm evaluation, we choose two metrics that are directly related to our objective function: feasibility ratio and average power per link. The feasibility ratio, κ, is the ratio of the number of feasible links $|\mathcal{L}_f|$ to the number of potential links $|\mathcal{L}|$:

$$\kappa = \frac{|\mathcal{L}_f|}{|\mathcal{L}|}. \tag{9}$$

Mobile nodes have limited battery life, and therefore power consumption is a concern for network longevity. We calculate the average power per link, χ, expressed as:

$$\chi = \frac{\sum_{i \in \mathcal{L}_f} P_i}{|\mathcal{L}_f|}. \tag{10}$$

5 Algorithm Evaluation

Our simulation environment, developed in MATLAB, considers L unidirectional links between transmitter–receiver node pairs in a frequency channel-based network. Transmitters are isotropic with unity gain and limited to a transmit power of $P_{\max}$. Every link experiences path loss, independent Rayleigh fading, and a noise floor of N_o. The network is assumed saturated, where each transmitter always has traffic to offer each receiver. Path loss is proportional to by $d_{i,j}^{-\alpha}$, where $d_{i,j}$ is the distance between the transmitter of link i and the receiver of link j and α is the path loss factor.

L links are randomly placed in a square simulation area under the constraint $d_{i,i} \leq d_{\max}$. C frequency channels are fixed, and each of the five algorithms discussed in Sect. 3 is executed to solve the DCPA problem. New topologies are generated and the algorithms are executed again until 1,000 trials are completed. Upon completion, metrics are collected and the simulation area is reduced. New trials are executed in a smaller simulation area. The simulation begins with a minimum density $D_{\min} = 10^{-5}$ links/m^2 and ends with a maximum density $D_{\max} = 10^{-1}$ links/m^2. In our simulation, we use parameter values of $\beta = 10$ dB, $N_o = -110$ dBm, $P_{\max} = 30$ dBm, $\alpha = 4$, and $L = 100$ links.

Our results show that the algorithms exhibit distinct regions of operation based on link density and the value $d_{\max}$. Figure 3 shows the performance comparison of the feasibility ratio and average power per link for $d_{\max} = 350$ (top plots) and $d_{\max} = 4000$ (bottom plots), for all algorithms. In Figure 3, the four regions of interest are labeled: noise dominant, transition, interference dominant, and path loss. The 90% confidence interval for the feasibility ratio is approximately ± 0.003 (left plots) and ± 2 dBm for average power per link (right plots).

The feasibility ratio plot with $d_{\max} = 350$ (top left), identifies the noise dominant region as the region where the feasibility ratio is 1 for the algorithms. In this noise dominant region, shorter link lengths, relative to the simulation area, minimize the effects of co-channel interference and produce a high feasibility ratio. Increasing link density moves algorithm performance into the transition region between the noise and interference dominant regions. In Fig. 3 (bottom left), the feasibility ratio declines when transitioning from the interference dominant region to the path loss region, because link density is low and $d_{\max}$ is large.

The average power per link plots provide corroborating data with the feasibility ratio plots. In the average power per link plot for $d_{\max} = 350$ (Figure 3, top right), as density increases, the average power per link initially rises and then slightly decreases or remains constant. In the noise dominant region, the SINR is minimally affected by co-channel links, allowing for slightly lower transmit power before the peak. However, as the density increases and the effects of co-channel interference become greater, links must compensate for this by increasing power. This power increase corresponds to the initial rise in power we see in Fig. 3 (top right). As the simulation transitions into the interference dominant region, the potential effects of co-channel interference reach their maximum. As density increases in the interference dominant region, algorithms show a slight decrease in transmit power because of the shorter link distances. Additionally, in the path loss region in Fig. 3 (bottom right) we see much higher power per link due to increased link lengths.

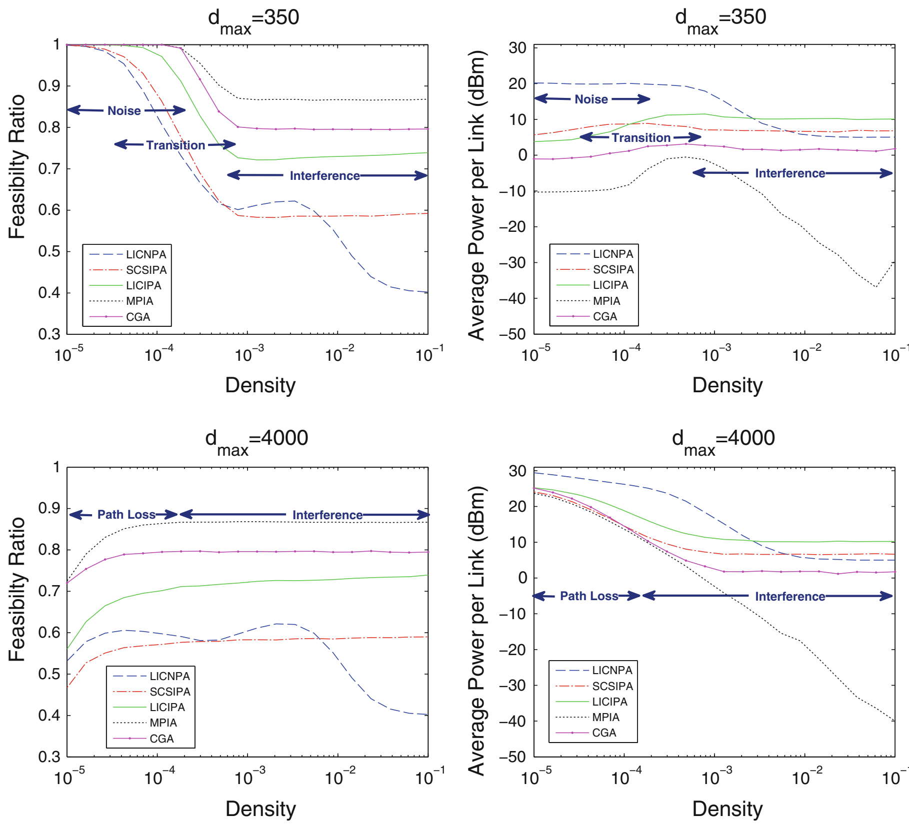


Fig. 3 Algorithm comparison using metrics feasibility ratio (left plots) and average power per link (right plots) with $L = 100$, $C = 40$ and with $d_{\max} = 350$ (top plots) and $d_{\max} = 4000$ (bottom plots). Plots with $d_{\max} = 350$ show a transition between a noise dominant and an interference dominant region. Plots with $d_{\max} = 4000$ show a transition between path loss and interference dominant regions. MPIA exhibits best feasibility ratio and requires the lowest transmit power

The results for LICNPA show distinct characteristics when compared to the other algorithms. These characteristics, most pronounced in the interference dominant region, are caused by the power control mechanism employed by the algorithm. LICNPA uses the quantity $\frac{I_{\text{th}}}{I_i}$ to adjust transmit power, which tends to excessively increase transmitter power and co-channel interference. As a result, LICNPA has the lowest feasibility ratio in the noise dominant and transition regions (Fig. 3, top left). In the interference dominant region, the quantity $\frac{I_{\text{th}}}{I_i}$ will be closer to one, thus there is a reduction in relative transmit power and an increase in feasibility ratio. This corresponds to the reduced power per link in Fig. 3 (top right) and the hump in the feasibility ratio in Fig. 3 (top left) around 10^{-3} links/m^2. As the density increases, LICNPA can only support as many available links as it has channels, since transmit power never reduces from its initial power of $P_{\text{ref}} = 5\,\text{dBm}$. The average power per link for LICNPA converges to 5 dBm in Fig. 3 (top right). In summary, the power control mechanism used by LICNPA can cause undesirable effects in link feasibility due to unnecessary transmit power.

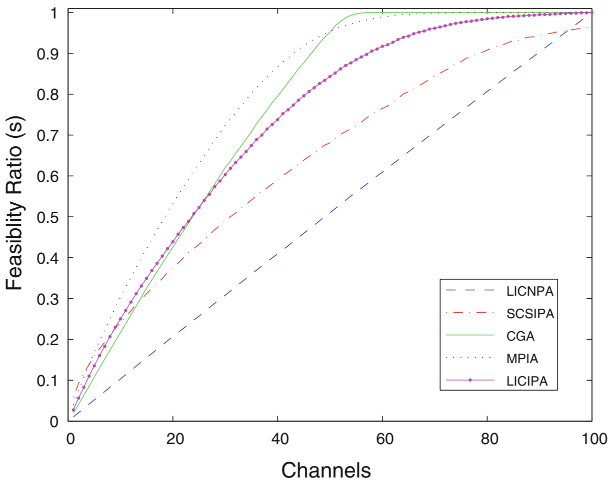


Fig. 4 Feasibility ratio as a function of the number of channels, with $L = 100$, and fixed density 10^{-1} links/m^2

In terms of relative algorithm performance, MPIA achieves the best feasibility ratio performance and the overall lowest average power per link. While MPIA appears to have all the desirable features of a DCPA algorithm, it also has a unique disadvantage. In MPIA, the inverse of the cross-link gain matrix is used to calculate ΔP_c. In some cases, this matrix is close to singular, resulting in an incorrect transmit power assignment. These near singular matrices occur when the link gain is much greater than the gain from co-channel interfering links, $G_{i,i} \gg G_{j,i}$. An example of the effects of this matrix approaching singularity can be seen by the "kink" shown in the Fig. 3 (top right). This "kink" is a result of incorrect power assignment, thus creating a wider confidence interval for average power per link.

Figure 4 shows, at a fixed density of 10^{-1} links/m^2 and $d_{\max}$ = 350, the feasibility ratio achieved by each algorithm as a function of the number of channels. In Fig. 4, performance differences are small for $C < 20$. For instance, LICIPA (a distributed algorithm) has almost identical performance as CGA (a centralized algorithm). When $C > 20$, algorithms show larger differences in feasibility ratio. Also, in some of the curves, the addition of more channels does not produce a linear improvement in feasibility ratio.

In addition to the complexity derivations in Sect. 3, we also compare the algorithms according to average run time. Average algorithm run time metrics are shown in Fig. 5 for a link density of 10^{-1} links/m^2, as a function of the number of channels. Three differences from our complexity analysis are apparent from Fig. 5. First, we would expect MPIA to have the highest run time because its complexity is estimated as $O(CL^4)$. However, in general, the matrix used to test the feasibility and calculate ΔP_c has smaller dimensions than $L \times L$. Therefore, the complexity analysis is overly pessimistic when compared to the average.

Two other differences involve the algorithms that use iterative power assignment: SCSIPA and LICIPA. The worst-case complexity for SCSIPA is $O(L^2)$, because we assume that $L \gg C$. When C increases, the number of receivers in the neighborhood of $d_{i,i}$ is likely to be fewer than C. Therefore, we expect the operation to determine all the available channels to dominate, and the run time to be closer to $O(LC)$. Additionally, SCSIPA assigns channels that are not being used by its $d_{i,i}$ neighbors and is more likely to iterate power on more links

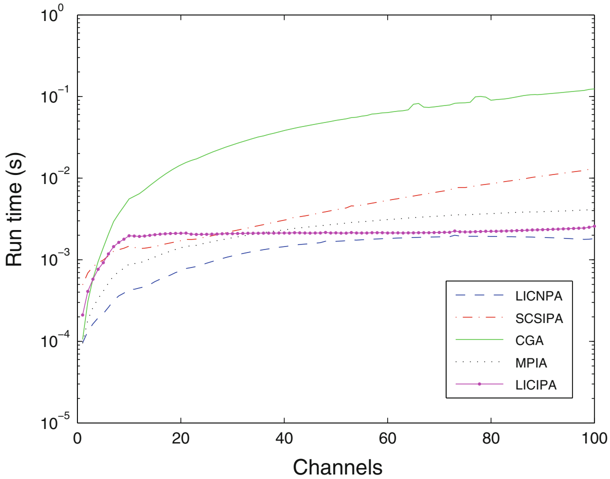


Fig. 5 Average run time shown as a function of C

than LICIPA. LICIPA uses the LIC for assignment and is more likely to affect fewer links. As a result, the average run time for SCSIPA is greater than for LICIPA as C increases.

6 Conclusions and Future Work

Our results demonstrate that for most of the channel assignment algorithms analyzed, the feasibility ratio is constant when in the interference dominant region. Additionally, we showed that comparative performance is a function of the number of channels and the link density. With few channels, the performance of all algorithms is comparable. Therefore, when considering network implementation with limited channel resources, the algorithm with the lowest complexity or run time should be selected.

The centralized algorithm MPIA has the overall best feasibility ratio and the lowest average power per link. Through assignment by the least change in transmit power, MPIA minimizes interference and increases the number of feasible links. However, implementation of this assignment faces the problem of near singular matrices. CGA showed comparable performance, with the disadvantage of a significantly longer run time. Additionally, in the case of the centralized algorithms, the requirement of knowledge of cross-link gains for any algorithm is problematic in the implementation of a real system. As a distributed alternative, LICIPA is a reasonable option.

LICIPA does not require cross-link gain information and exhibits the overall lowest run time, complexity, and the overall best feasibility ratio among the distributed algorithms. As a comparison, when CGA reaches a feasibility ratio of 1 at $C \approx 60$, LICIPA has a feasibility ratio of ≈ 0.9 (Fig. 4). However, this does come at a cost of slightly higher average power per link. Compared to SCSIPA, LICIPA does not require location and channel information for assignment. Instead, it measures received power on all channels and selects the LIC. In terms of power iterations, LICIPA will cause fewer perturbations in overall network power than SCSIPA, because LICIPA uses the LIC. However, measuring received power on all channels does come at some computational cost, whereas selecting a channel based on control messages may incur less computational cost but generate more network overhead.

While channel assignment has been a widely explored field, we believe some open issues remain. First, how should a network dynamically adapt based on changing conditions due to spectrum availability and desired quality of service? We believe this is important when considering the case of primary and secondary spectrum users. Second, how can a network maintain an optimal global topology through channel assignment when nodes only have access to local information? While some work has been done in combining aspects of routing and channel assignment, we believe there is still opportunity for contribution in this area. Finally, what learning and self-organization techniques could a CN use to improve network performance and what would be the tradeoffs? Overall, we believe that the analysis of channel assignment is an important first step in understanding how to create CN.

Acknowledgements We would like to thank G. Kulkarni for providing us with his MATLAB code to develop the algorithm MPIA.

References

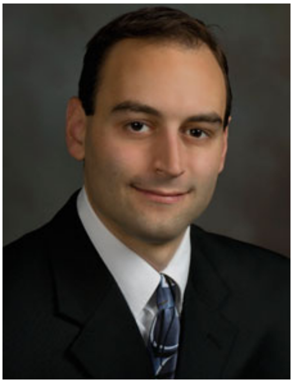
1. Chin, C.M., Sim, M.L., Olafsson, S. (2007). A dynamic channel assignment strategy via power control for ad-hoc network systems. In *Vehicular Technology Conference, 2007 VTC2007-Spring IEEE 65th pp. 3006–3010*. doi:10.1109/VETECS.2007.616.
2. ElBatt, T., & Ephremides, A. (2004). Joint scheduling and power control for wireless ad hoc networks. *IEEE Transactions on Wireless Communications, 3*(1), 74–85.
3. Foschini, G., & Miljanic, Z. (1993). A simple distributed autonomous power control algorithm and itsconvergence. *IEEE Transactions on Vehicular Technology, 42*(4), 641–646.
4. Grace, D., Tozer, T., & Burr, A. (2000). Reducing call dropping in distributed dynamic channel assignment algorithms by incorporating power control in wireless ad hoc networks. *IEEE Journal on Selected Areas in Communication, 18*(11), 2417–2428.
5. Kulkarni, G., & Srivastava, M. (2002). A channel assignment scheme for FDMA based wireless ad hoc networks in Rayleigh fading environments. In *2002 IEEE 56th Vehicular Technology Conf. (VTC 2002-Fall), vol 2*.
6. Skiena, S. (1998). *The algorithm design manual*. Berlin: Springer.
7. Thomas, R., Friend, D., DaSilva, L., & MacKenzie, A. (2006). Cognitive networks: Adaptation and learning to achieve end-to-end performance objectives. *IEEE Communications Magazine, 44*(12), 51–57.
8. Zander, J. (1992). Performance of optimum transmitter power control in cellular radiosystems. *IEEE Transactions on Vehicular Technology, 41*(1), 57–62.

Author Biographies

Juan D. Deaton is currently employed as a Cellular Systems Engineer at the Idaho National Lab and a Ph.D. Student at Virginia Tech. Before becoming a graduate student, Juan's work involved researching options for wireless airborne emergency communications and vulnerabilities of VoIP applications. At Virginia Tech, his focus is in wireless cognitive networking under the advisement of Dr. Luiz DaSilva.

Syed A. Ahmad completed his Bachelors in Computer Engineering in 2005 from Lahore University of Management Sciences (LUMS), Pakistan. In 2007, he completed his Masters in Electrical Engineering from West Virginia University, Morgantown. Since August 2007, he has been a Ph.D. student at Virginia Tech where the focus of his research is on how cross-layer adaptations impact the performance of cognitive networks. His research interests include wireless networks, adaptive rate schemes, spread spectrum communication and multiple antenna systems.

Umesh Shukla is pursuing his masters degree in Electrical & Computer Engineering Department at Virginia Tech. Before coming to Virginia Tech, he worked as a software developer in India for one year. He received his bachelors degree from National Institute of Technology, Allahabad, India. His primary research interest is in resource allocation techniques for wireless networks. He is a member of Wireless@VT research group.



Ryan E. Irwin has been a graduate student at Virginia Tech since the fall of 2007. Within the ECE department he is part of the Wireless @Virginia Tech research group. His focus is in wireless networking and communication under the co-advisement of Dr. Allen MacKenzie and Dr. Luiz DaSilva. Ryan obtained a B.S. in computer engineering from Mississippi State University in May of 2007.

Luiz A. DaSilva is currently on leave from Virginia Tech's Bradley Department of Electrical and Computer Engineering, where he has been a faculty member since 1998. He currently holds the Stokes Professorship in Telecommunications in the Department of Electronic and Electrical Engineering at Trinity College Dublin, in Ireland. Prof. DaSilva's research focuses on distributed and adaptive resource management in wireless networks, and in particular cognitive radio networks and the application of game theory to wireless networks. He is currently a principal investigator on research projects funded by the National Science Foundation, DARPA, and the European Commission under Framework Programme 7. Prof. DaSilva is a Senior Member of IEEE and a member of the ASEE and of ACM. In 2006, he was named a College of Engineering Faculty Fellow at Virginia Tech.

Allen B. MacKenzie has been an Assistant Professor in Virginia Tech Bradley Department of Electrical and Computer Engineering since 2003. He joined Virginia Tech after receiving his Ph.D. from Cornell University and his B.Eng. from Vanderbilt University, both in Electrical Engineering. Dr. MacKenzie's research focuses on wireless communications systems and networks. His current research interests include cognitive radio and cognitive network algorithms, architectures, and protocols and the analysis of such systems and networks using game theory. Dr. MacKenzie is a senior member of the IEEE and a member of the ASEE and the ACM. In 2006, he received the Dean's Award for Outstanding New Assistant Professor in the College of Engineering at Virginia Tech.