Proceedings of the 45th IEEE Conference on Decision & Control
Manchester Grand Hyatt Hotel
San Diego, CA, USA, December 13-15, 2006

ThA18.3

Convergence of Distributed Learning Algorithms for Optimal Wireless Channel Allocation

D.J. Leith, P. Clifford
Hamilton Institute, National University of Ireland, Maynooth, Ireland
{doug.leith, peter.clifford}@nuim.ie

Abstract— In this paper we establish the convergence to an optimal non-interfering channel allocation of a class of distributed stochastic algorithms. We illustrate the application of this result via (i) a communication-free distributed learning strategy for wireless channel allocation and (ii) a distributed learning strategy that can opportunistically exploit communication between nodes to improve convergence speed while retaining guaranteed convergence in the absence of communication.

I. INTRODUCTION

In this paper we consider the problem of allocating radio channels in a wireless network. The channel allocation task has been the subject of a considerable literature, spanning cellular networks (e.g. see the survey paper [1] and [2], [3], [4], [5]), wireless LANs (e.g. see [6], [7], [8], [9], [10], [11] and references therein) and graph theory (our channel allocation task is equivalent to the classical graph-colouring problem [12], [13]). Almost all of this work has, however, been concerned either with centralised schemes or with distributed schemes that rely upon extensive message-passing. In this paper we consider distributed channel allocation strategies that do not require any communication or message-passing.

Figure 1 shows an example of interfering wireless networks (WLANs). Transmissions within the AP1 and AP2 WLANs can interfere, with the interference range of each WLAN indicated by the dashed circles (the use of circles is purely indicative – interference domains may be much more complex depending upon the distribution of client nodes, physical obstacles, etc). The level of interference between any particular pair of transmissions depends on the physical locations of the communicating stations. The channel selection problem considered here is equivalent to graph colouring. To see this, define the interference graph by associating a node with each WLAN (e.g. with each BSS in an 802.11 network) and inserting an edge between nodes that interfere. For example, Figure 2 shows the interference graph corresponding to the wireless network in Figure 1. A colouring of the graph assigns colours to each node, and a proper colouring is an assignment of colours to each node such that no adjacent nodes share the same colour. A non-interfering channel allocation is thus equivalent to a proper colouring of the interference graph associated with a wireless network. In this simple example the optimal channel selection itself is straightforward to see, but this need not
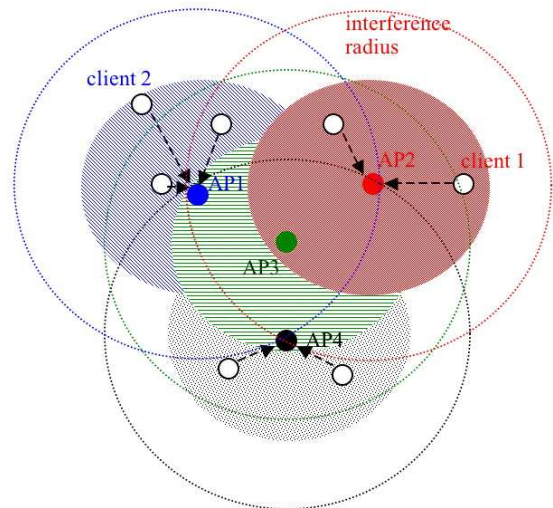

Fig. 1. Example of interfering 802.11 WLANs. Dashed circles indicate interference radius, shaded circles indicate communication radius.
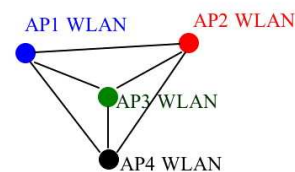

Fig. 2. Interference graph of Figure 1.

be the case in more complex topologies. Indeed, since the underlying channel selection problem considered here is equivalent to graph colouring it is thus known to be NP-hard.

Our main result in this paper is to establish the convergence to an optimal non-interfering channel allocation of a class of distributed stochastic algorithms. The result is quite general and creates a broad framework within which the design of efficient algorithms with guaranteed convergence can be studied. We illustrate the application of this result via (i) a communication-free distributed learning strategy for wireless channel allocation and (ii) a distributed learning strategy that can opportunistically exploit communication between nodes to improve convergence speed while retaining

guaranteed convergence in the absence of communication.

II. MAIN RESULT

We are interested in establishing a class of distributed algorithms with guaranteed convergence to an optimal non-interfering channel allocation. Let $G = (V, E)$ denote the interference graph associated with a wireless network. V denotes the set of graph nodes and E the edges. The nodes do not know the interference graph. Let $C(V_i) \in \{1, 2, ..., c\}$ denote the channel used by node V_i and $N(V_i)$ denote the set of neighbours. We begin by defining "success" at node V_i to be when node V_i selects a different channel from all of its neighbours i.e. when $C(V_i) \cap \{C(V_j) : V_j \in N(V_i)\} = \emptyset$, and have "failure" at node V_i otherwise. We allow determination of the "success" or "failure" of a channel selection at node V_i to take sensing time $T(V_i) \geq 0$. For example, in a wireless context this might reflect the time to sense channel quality using a packet error rate measure derived from multiple packet transmissions on the selected channel. We require that there exists an upper bound $T(V_i) \leq T_{max}$ $\forall V_i \in V$ but otherwise place no restrictions on the sensing times. In particular, we do not require sensing events to be synchronised.

We consider distributed algorithms where each node updates its channel selection after sensing "success" or "failure" with the current choice of channel. Each node also maintains and updates channel selection probabilities in order to allow learning. We consider the class of algorithms with the following two properties:

1) When a channel selection yields "success" at node V_i, that node continues to use the same channel. Note that this guarantees that any channel allocation which removes interference between all nodes is an absorbing state.

2) When a channel selection yields "failure", *every other* channel has at least probability $pr_{lb} > 0$ of being chosen next.

We refer to such algorithms as Distributed Learning (DL) algorithms.

Theorem 1 *Suppose each node in a graph G operates a DL algorithm. Assume that the channel allocation problem is feasible (i.e. the number of available channels c is greater than or equal to the chromatic number χ of G). Then the algorithm converges, with probability one, to an optimal channel allocation.*

We will show that in a determined finite amount of steps the system has some minimum positive probability of convergence. In the following we refer to the collection of channel selections and channel selection probabilities as the configuration of the system. We show that starting from any configuration the system can reach some standard state after two steps. From this standard state we show that the system can then potentially reach a state where every node experiences a failure simultaneously, allowing convergence without issues of dependence between node states. Hence

the network always has lower bounded probability of global success and so will almost surely converge.

In the sequel we refer to two or more nodes choosing the same channel as a "collision". We say that the state $\mathbb{S}$ consists of all possible unsuccessful configurations with all colour selection probabilities of nodes which have just collided strictly greater than pr_{lb}. Note that being in state $\mathbb{S}$ requires that any channel just collided on has probability at least pr_{lb} of being chosen again, a stronger condition than Property 2. Denote the maximum node degree by md and the diameter of the graph (length of the longest shortest path between two nodes) by D.

The system may avoid state $\mathbb{S}$ by some node undergoing repeated same channel collisions. Property 2 of our DL algorithm is required to prevent this happening. We show in Lemma 1 that if the system has reached a configuration with some colour selection probabilities very small there is a positive lower bounded probability that it will return in two steps to our standard state $\mathbb{S}$. Thus (with some probability) the initial colour selection probabilities will have no effect on the probability of a given evolution.

Lemma 1. From any configuration of the system, if after two steps the system has not converged, the system is in state $\mathbb{S}$ with some probability $pr_5 > 0$.

Proof of Lemma 1. After any step T_0 there was either global success (and convergence) or at least two nodes suffered a collision. Starting at time T_0 we allow the system to evolve for 2 more steps and lower bound the probability of the system not being in state $\mathbb{S}$. We ignore nodes who succeed and then collide as their colour selection probabilities are clearly at least pr_{lb}. Consider any just collided node; after the collision on colour i_1, it has probability $pr_1 > pr_{lb}$ of choosing some specific other colour i_2 and probability $pr_2 > (c-1)pr_1$ of choosing any colour other than i_1. So the probability of two repeated collisions on the same colour at a specific node is $pr_3 < 1 - pr_2$. In the whole system the probability of some node having two consecutive same colour collisions is $pr_4 < npr_3 - \binom{n}{2}(pr_3)^2 + \cdots < 1$. Hence with some probability $pr_5 > 1 - pr_4 > 0$ the system has no node with consecutive same colour collisions. Thus after these two steps with probability pr_5 all colour selection probabilities of nodes which have just collided are strictly greater than pr_{lb} and the system is in state $\mathbb{S}$. ∎

Lemma 2. Suppose that the system is in state $\mathbb{S}$. Assume without loss of generality that the graph is connected. There exists a specific evolution $\mathbb{E}$ of the system which results in all nodes transmitting succesfully.

Proof of Lemma 2. In state $\mathbb{S}$ by definition we have not converged and (at least) two nodes k_1 and k_2, say, have just experienced a collision. By way of notational convenience we say these two nodes were *visited* at step 2. Suppose now that k_1 collides with its first non visited neighbour k_3 (if any) at step 3. Suppose also that k_2 collides with its first non visited neighbour (if any, potentially k_3 also) at step 3 also. We say that such nodes are *visited* at step 3. Inductively suppose now that a node once visited collides with all its nonvisited neighbours in consecutive steps. This is possible because a

visited node having just collided can potentially choose any colour. Note that a node being visited simultaneously (along two different equal length paths from k_1 and k_2 say) is also possible.

Suppose that once a node has collided with all its non-visited neighbours it then repeatedly chooses colour 1 until step $T_1 = T_0 + 3 + md \times D$. We note that as a node k_4 is colliding with its nonvisited neighbours some of them may become visited from other nodes before they collide with k_4; we suppose then that k_4 does not visit such nodes. Note also that $md \times D$ upper bounds the time needed for this visiting procedure to visit every node.

By connectedness, at time $T_1 - 1$ it is possible for every node to have been visited and to be choosing colour 1. Hence every node is colliding. Since every node is colliding, every node can choose every colour, so we can finally suppose that at step T_1 every node selects a colour so that no collisions occur. ∎

Lemma 3. There is a strictly positive lower bound pr_8 on the probability of the evolution $\mathbb{E}$ occurring from any configuration in state $\mathbb{S}$.

Proof of Lemma 3. Given the initial colour selection probabilities and which nodes collided initially, the evolution $\mathbb{E}$ is well defined and has some positive (computable) probability pr_6 of occurring since the system is finite.

By assumption the system begins in state $\mathbb{S}$ and so the initial colour selection probabilites of just collided nodes are lower bounded; therefore there is some probability $pr_7 > 0$ such that $pr_6 > pr_7$ irrespective of the initial colour selection probabilities.

The subset of nodes which collided initially is one of finitely many possibilities and so again there is some probability $pr_8 > 0$ such that $pr_7 > pr_8$ irrespective of which nodes collided initially. ∎

Proof of Theorem 1. Defining $pr_9 = pr_8 pr_5$ gives the probability that the system is in state $\mathbb{S}$ after the first two steps and then follows evolution $\mathbb{E}$. Hence every $2 + md \times D$ steps the system will converge with probability at least pr_9. Hence after $j(2 + md \times D)$ steps we have converged with probability at least $1 - (1 - pr_9)^j$ which converges to 1 as $j \to \infty$. ∎

This proof actually provides a partial answer to a further question, namely how quickly the algorithm converges to an optimal allocation. We have the following property.

Corollary 1 *Let τ denote the stopping time of the DL algorithm. Then $prob[\tau > k] < \alpha e^{-\gamma k}$, for positive constants α, γ.*

That is, the stopping time probability decays exponentially. Our argument does not yield a tight estimate of the exponent γ, which determines the precise convergence rate of the algorithm, but given that the underlying colouring problem is NP-hard this is unsurprising.

III. APPLICATIONS

A. Distributed Learning Automata

Our main result establishes the convergence of a class of distributed channel allocation algorithms. We first note that this class includes interconnected learning automata. Specifically, let c denote the number of available channels and let each access point with responsibility for channel selection maintain a c element state vector p. Let p_i denote the ith element of p with $\sum_i^c p_i = 1$. Consider the following distributed learning algorithm for updating p at each node.

Communication Free Learning (CFL) Algorithm
1) Initialise $p = [1/c, 1/c, \ldots, 1/c]$
2) Toss a weighted coin to select a channel, with p_i the probability of selecting channel i. Sense the channel quality/transmit.
3) On a successful choice of channel i, update p as

$$p_i = 1, p_j = 0 \ \ \forall j \neq i \qquad (1)$$

i.e. on a successful choice we use the same channel for the next round.
4) On failure on channel i, update p as

$$p_i = (1-b)p_i, p_j = (1-b)p_j + \frac{b}{c-1} \ \ \forall j \neq i \quad (2)$$

i.e. on a failure multiplicatively decrease the probability of using that channel, redistributing the probability evenly across the other channels. b is a design parameter, $0 < b < 1$.
5) Return to 2.

We observe that the CFL algorithm employs a linear reward - penalty learning automata [14] at each node. These automata are coupled via the interference graph but do not otherwise communicate or use message-passing (hence why we describe this as a communication-free learning algorithm). It can be seen immediately that the CFL algorithm satisfies Properties 1 and 2 (with $pr_{lb} = \frac{b}{c-1}$). It therefore follows from Theorem 1 that with probability one the CFL algorithm converges exponentially to an optimal non-interfering channel allocation. For further information on this CFL algorithm such as optimal values for b see [15].

We note that previous work on learning automata has largely focussed on individual automata rather than the interconnection of a large number of automata, with few previous results known about the properties of interconnected learning automata. To our knowledge, Theorem 1 is one of the first convergence results for interconnected learning automata.

B. Opportunistic Partial Communication

The foregoing CFL algorithm is unusual in that it requires no communication or message-passing between nodes. As noted previously, the wireless channel allocation task has been the subject of a considerable literature, spanning cellular networks (e.g. see the survey paper [1]), wireless LANs (e.g. see [7], [8], [9], [10], [11] and references therein) and

2982

graph theory. Almost all of this work has, however, been concerned either with centralised schemes or with distributed schemes that rely upon extensive message-passing.

The requirement for message passing means in such schemes is clearly problematic in many realistic contexts where interfering networks may belong to different administrative domains (e.g. interfering wireless networks may be operated by different households or businesses), while packet sniffing on the radio channel runs into the difficulties that packets may be encrypted and/or the distance over which packets are readable is typically much less than the distance over which network transmissions interfere (thus interfering access points may well not be able to sniff each others packets). Moreover, most of the proposed distributed schemes are heuristic in nature and come with few performance guarantees (partly due to the NP-hard nature of the channel allocation problem, although NP-hardness only relates to the computational complexity of the problem). A distributed channel allocation strategy, such as the CFL algorithm, with guaranteed convergence that does not depend upon communication or message-passing is therefore potentially very attractive from a practical viewpoint.

Of course, while we do not want to rely upon communication, when communication *is* possible we would like our channel allocation algorithm to take advantage of this feature as we would expect it to increase performance e.g. reduce convergence time. We therefore consider extending the basic CFL algorithm to allow it to opportunistically take advantage of packet sniffing on the radio channel when this is possible. We note that channel conditions can be expected to vary stochastically and transmissions from neighbouring may be intermittent. Moreover, the distance over which transmissions interfere (which defines the neighbours of a node) is generally greater than the distance over which transmissions can be decoded (which determines the maximum communicating set of neighbours). Let $N_c(V_i,t)\} \subseteq N(V_i)$ denote the set of neighbours for which node V_i is able to decode packet transmissions at round t of operation. We have a situation where the membership of $N_c(V_i,t)$ varies stochastically in time and where allowable values of $N_c(V_i,t)$ include the empty set. It is important to emphasise that this opportunistic partial communication scenario is fundamentally different from the communication models assumed in most other work on distributed algorithms for channel allocation. In previous work, algorithms require local communication between *all* nodes in an interference neighbourhood or perhaps all nodes in a k-hop interference neighbourhood. That is, $N_c(V_i,t)\} \subset N(V_i)$ is inadmissible and it is strictly required that that $N_c(V_i,t)\} \supseteq N(V_i)$.

We consider the following opportunistic communication extension to the basic CFL algorithm.

Opportunistic Learning Algorithm
1) Initialise $p = [1/c, 1/c, \ldots, 1/c]$
2) Toss a weighted coin to select a channel, with p_i the probability of selecting channel i. Let C denote the outcome of the coin toss i.e. the channel selected.
3) Broadcast the channel selected in a small beacon

packet. [1]
4) Receive any neighbouring broadcasts and opportunistically record their planned channel selection. Let C_{N_c} denote this set of planned channel selections.
5) If $C \cap C_{N_c} = \emptyset$, then do nothing. Otherwise, we know that our choice C of channel will lead to a failure (a neighbour will choose the same channel) and so we revisit our channel selection. With probability $1 - \delta$ we choose a channel uniformly randomly from the set of channels not in C_{N_c} and with probability δ we stay with the original channel selection C.
6) Sense the channel quality / transmit.
7) On a successful choice of channel i, update p as

$$p_i = 1, \ p_j = 0 \ \ \forall j \neq i$$

i.e. on a successful choice we use the same channel for the next round.
8) On failure on channel i, update p as

$$p_i = (1-b)p_i, \ p_j = (1-b)p_j + \frac{b}{c-1} \ \ \forall j \neq i$$

i.e. on a failure multiplicatively decrease the probability of using that channel, redistributing the probability evenly across the other channels. b is a design parameter, $0 < b < 1$.
9) Return to 2.

This algorithm is identical to the CFL algorithm apart from the opportunistic communication steps 4 and 5. The modified algorithm clearly satisfies the absorbing property 1. Since there is always some probablity δ of following the original CFL algorithm, the channel selection probabilities are also lower bounded i.e. property 2 is also satisfied. It follows that convergence of the modified algorithm is guaranteed with probability one by Theorem 1. Only small changes are required to adapt this algorithm to the situation where the stations are not synchronised; the adapted algorithm is of course convergent also.

The benefit of opportunistically exploiting communication is illustrated in Figure 3. In this example the network interference graph is modeled as a random disk graph; that is, nodes are uniformly randomly located in a unit square and the WLANs associated with two nodes interfere when the nodes are located within a radius R of each other. A "failure" or "collision" occurs when neighbouring nodes select the same channel at a given iteration of the channel allocation algorithm, and a "success" when a node selects a different channel from all of its neighbours. For each interference graph the chromatic number χ gives the minimum possible number of channels for a feasible solution. We model communication between nodes as an embedded disk graph with radius $R_c \leq R$ i.e. a node can decode packet transmissions from neighbouring nodes that lie within communication radius R_c. Figure 3 shows the mean number of iterations for the network to convergence to an optimal non-interfering

---

[1]Any reasonable implementation suffices since communication with all nodes in an interference neighbourhood is not required.
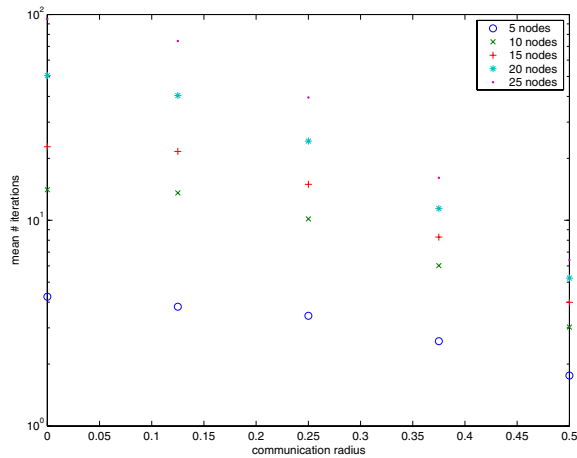
2983

Fig. 3. Mean number of iterations to converge to an optimal channel allocation vs communication radius in interference graph (random disk graphs with radius R=0.5, mean is taken over 1000 graphs, number of channels $c = 1.2\chi$, $b = 0.1$, $\delta = 0.1$)

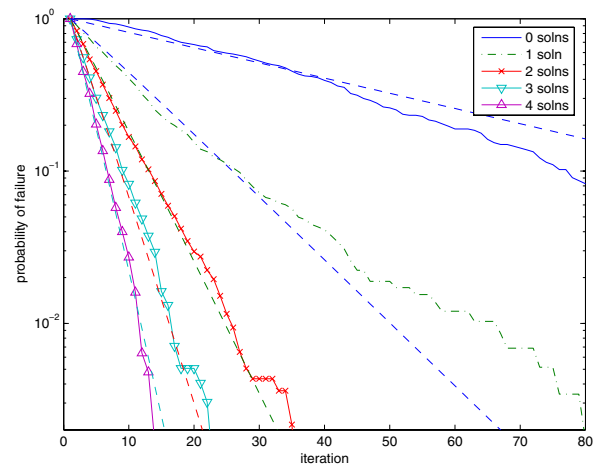Fig. 4. Probability of failure (1- prob of success) vs iteration following addition of a new node. Dashed lines are analytic predictions. "Solutions" refer to the number of possible channels that the new node may select to achieve a proper channel allocation without disturbing the allocations of the original nodes. (20 node random disk interference graph with R=0.5, number of channels c=12 (1.25χ), $b = 0.1$.)

channel allocation vs the communication radius and number of nodes. Note that a log scale is used on the y-axis. It can be seen that a significant improvement in provided by the communication step (performance with no communication at all is illustrated by the communication radius R_c being zero). The greatest gain in performance, roughly an order of magnitude reduction in convergence time, is achieved when $R_c = R$ i.e. when a node can communicate with all of its interfering neighbours. Interestingly, we can also see that the performance gain is strongly dependent on the level of communication. A significant proportion of node neighbours must be able to communicate before the most substantial performance gains are realised. In particular, we note that a common guideline is that the communication radius may be approximated as half the interference radius. This corresponds to $R_c = 0.25$ in Figure 3 since $R = 0.5$. Hence, for example, with 25 nodes the mean convergence time is 95 iterations when $R_c = 0$, 40 iterations when $R_c = 0.25$ and 6.5 iterations when $R_c = R = 0.5$.

We can gain some insight into the source of this improvement in convergence rate by considering an interference graph with an optimal channel allocation and adding a new node. For example, Figure 4 shows simulation results for a network with a random disk interference graph. In this example we take a network where nodes are located randomly in the plane and nodes interfere if they are within a distance R of one another. We then randomly[2] add a single new node and record the probability of success and the number of collisions that occur. We do this repeatedly (always starting from the same network and randomly adding one new node) to sample the distribution. We expect that the convergence time will depend upon the number of free

---

[2]The network is located in the plane, making it straightforward to add a new node. Specifically, we select uniformly random x and y co-ordinates for the new node and then determine its neighbours using the interference radius R.

channels in the neighbourhood that the new node could select to achieve a proper channel allocation. In Figure 4 we therefore bin the simulation data according to the number of local solutions.

Also shown in Figure 4 are the predictions corresponding to the following simple analysis. Let F_c denote the set of free channels initially that the new node might select to achieve a non-interfering channel allocation. Assume, for the moment, that F_c consists of only one channel. Let $p(k)$ denote the probability of the new node choosing this channel at iteration k. Assume that the channel allocation of nodes in the original network remains unchanged. Then on a collision p is updated according to

$$p(k + 1) = (1 - b)p(k) + b/(c - 1) \tag{3}$$

We have that $E[F(0)] = 1 - p(0)$ and $E[F(k + 1)] = E[F(k)](1 - p(k + 1))$ with $E[S(k)] = 1 - E[F(k)]$. Evidently, this argument can be directly generalised to cases when F_c contains more than one channel. This analysis can be applied provided there exists at least one local solution, and it can be seen from Figure 4 that in such situations it yields remarkably accurate predictions. This indicates that the channel allocations of the nodes in the original network possess sufficient "inertia" that they do indeed effectively remain unchanged (this is also confirmed by direct measurement of the network channel allocations before and after the addition of a new node). It follows immediately from this insight that partial communication with its neighbours enables the new node to select a free channel with higher probability, thereby accelerating convergence. Note that the convergence rate depends geometrically on the probability of selecting a free channel and thus very substantial performance gains are possible.

2984

It can of course happen that there exist *no* local solutions, i.e. all of the available channels are already used by the neighbours of the new node. This situation is marked as the "zero solutions" curve in Figure 4. In this case a non-local re-allocation of channels is necessary in order to achieve a non-interfering channel allocation and the previous analysis cannot be applied. It can be seen from Figure 4 that the convergence is this case is slower than when local solutions exist. This is, of course, to be expected as non-local re-allocation requires both at least one of the original nodes to change channel and for the new node to then select one of the resulting free channels.

We can carry out an approximate analysis of this case as follows. Denote the set of nodes neighbouring the new node by N. We know that these nodes make use of all available channels. Our measurements on many hundreds of thousands of disk graphs indicate that we almost never see adjacent nodes such that the neighbourhoods of *both* nodes make use of all available channels. We therefore assume that the neighbours N do themselves have the freedom to change channel. Consider the behaviour of the new node: because there is no local solution it must choose the same channel as one of its neighbours. By assumption, a neighbour will change channel with probability at least $b/(c-1)$ and otherwise stay on the same channel. Note that it can occur that more than one neighbour shares the same channel, in which case we need all such neighbours to change channel in order to free up that colour. This possibility is neglected in our model because simulations show it is a rare occurrence. Hence our model predicts that independently at each timestep, the system will reconverge approximately with probability at least $b/(c-1)$. The accuracy of this approximate analysis is illustrated in Figure 4. Again, it follows immediately from this analysis that the convergence rate can be made significantly faster when even partial communication between nodes is possible. Specifically, all nodes operate the opportunistic learning algorithm, including the neighbours N. The new node will necessarily induce failures at its neighbours N since no local solutions exist. Communication facilitates these neighbours selecting a channel that does not collide with the new node or their own neighbours, thereby freeing up channels in the neighbourhood of the new node and this accelerating convergence.

IV. Conclusions

In this paper we establish the convergence to an optimal non-interfering channel allocation of a class of dis-

tributed stochastic algorithms. The result is quite general and creates a broad framework within which the design of efficient algorithms with guaranteed convergence can be studied. We illustrate the application of this result via (i) a communication-free distributed learning strategy for wireless channel allocation and (ii) a distributed learning strategy that can opportunistically exploit communication between nodes to improve convergence speed while retaining guaranteed convergence in the absence of communication.

V. Acknowledgements

References

[1] L. Narayanan, "Channel assignment and graph multicoloring," *Handbook of wireless networks and mobile computing*, Wiley series on parallel and distributed computing, 2002.
[2] P. Sparl, J. Zerovnik, "2-local 5/4-competitive algorithm for multicoloring triangle-free hexagonal graphs", *Information Processing Letters* 90, 2004.
[3] K. S. Sudeep, S. Vishwanathan, "A technique for multicoloring triangle-free hexagonal graphs",' *Discrete Mathematics* 2002.
[4] J. Janssen, D. Krizanc, L. Narayanan, S. M. Shende, "Distributed on-line frequency assignment in cellular networks," *Proc. of the 15th annual symposium on theoretical aspects of computer science*, Lecture Notes in Computer Science Vol. 1373, 1998.
[5] L. Narayanan, S. M. Shende, "Static frequency assignment in cellular networks", *Algorithmica* 29 (2001).
[6] B. Kauffmann, F. Baccelli, A. Chaintreau, K. Papagiannaki, C. Diot, "Self Organization of Interfering 802.11 Wireless Access Networks,", *INRIA Technical Report*, August 2005.
[7] L. Tassiulas, A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks", *IEEE Trans Automatic Control*, 37 (12), 1992.
[8] A. Raniwala, T. Chiueh. "Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network". In *Proc IEEE International Conference on Computer Communications*, 2005.
[9] H. Luo, P. Medvedev, J. Cheng, S. Lu, "A self coordinating approach to distributed fair queuing in ad hoc wireless networks", *Proc. of IEEE INFOCOM '01*, 2001.
[10] B.J. Leung, K.K. Kim, "Frequency assignment for IEEE 802.11 wireless networks". *Proc. 58th IEEE Vehicular Technology Conference*, 2003.
[11] Y. Bejerano, S.-J. Han and L. (Erran) Li, "Fairness and load balancing in wireless LANs using association control", *MobiCom '04*, 2004.
[12] I. Finocchi, A. Panconesi, R. Silvestri, "Experimental analysis of simple, distributed node coloring algorithms", *Algorithmica* 41 (2005).
[13] M. Kubale, L. Kuszner, "A better practical algorithm for distributed graph coloring," *Proc. of IEEE PARELEC'02*, 2002.
[14] K. Narendra, M. A. L. Thathachar, "Learning Automata: An Introduction", Prentice Hall, 1989.
[15] D. Leith, P. Clifford, "A Self-Managed Distributed Channel Selection Algorithm for WLANs". In *Proc IEEE RAWNET '06*, 2006.