CONVERGENCE OF A QUASISTATIC FREQUENCY ALLOCATION ALGORITHM

BART PRENEEL and JEAN WALRAND *

Department of EECS
UNIVERSITY OF CALIFORNIA
BERKELEY, CA 94720
{preneel,wlr}@eecs.berkeley.edu

August 8, 1995

Abstract

The paper proves the convergence of a quasistatic frequency allocation algorithm for a cellular network. The algorithm is distributed and asynchronous and is executed by the base stations at random epochs when they sense a conflict with other transmitters. We prove that the algorithm eventually finds an acceptable allocation of frequencies whenever one exists. We also derive bounds on the average number of steps required by the algorithm. We study methods for speeding up the convergence, and derive analytical expressions for the case where the base stations are located on a line. We present the results of computer simulations, which show a very fast convergence.

Key words: cellular radio, frequency allocation, frequency division multiple access, graph coloring

1 Introduction

This research is motivated by a problem that arises when operating a cellular network. The network consists of N base stations that operate on a number $F < N$ of distinct frequencies. The problem that we analyze is that of allocating frequencies to the base stations. A given allocation of frequencies is *acceptable* if base stations that use the same frequency are far enough apart for their interference to be small enough. See for instance [7].

In this paper we analyze an algorithm, similar to the algorithm discussed in [1], that lets the base stations select their operating frequencies automatically. We describe

*Research supported in part by DIVA COMMUNICATIONS. The first author is a postdoctoral researcher, sponsored by the N.F.W.O. (National Science Foundation of Belgium).

the algorithm and show that it eventually finds an acceptable frequency allocation if one exists. Such an automatic algorithm adapts to changing operating conditions created by the addition of new base stations or by the activity of an unknown transmitter. We call the algorithm *quasistatic* because we assume that the external conditions do not change while the base stations execute the algorithm.

A different approach is to use dynamic channel assignment (DCA), where all frequencies are available to all base stations, and the frequency is assigned on a call-by-call basis [5]. A distributed algorithm for DCA is described in [2], and in [4] a class of algorithms is defined which is asymptotically optimal in the sense that for increasing offered traffic, the carried traffic approaches the maximum possible value. A comparison of the dynamic and quasistatic approaches can be found in [3]. The main conclusion is that quasistatic frequency assignment is more suitable for systems with heavy traffic loads.

We discuss our mathematical model of the cellular network in Section 2. In particular, that section specifies what we mean by acceptable frequency allocation. Section 3 discusses two distributed frequency allocation algorithms and explains why the first one, which may be appealing to implementors, may not converge to an acceptable frequency allocation. Section 4 proves the convergence of the second algorithm. In Section 5 we present some results on the time to convergence of the algorithm. The general bounds are overly pessimistic, but they are improved by taking into account some heuristic information on the problem. In Section 6 we discuss methods to speed up the convergence. This includes a parallelization of the algorithm. Section 7 presents analytical results for the case where the base stations are lying at equal distance on a line. In Section 8 we discuss the results of extensive simulations. Finally, some concluding remarks are presented in Section 9.

2 Mathematical Model of the Cellular Network

We are concerned with the evolution in time of the frequency allocation. Consequently, we do not need to model the details of the operations of the cellular network.

Our cellular network has N base stations. Each base station uses one of F frequencies. A *frequency allocation* is an element of $\{1, \dots, F\}^N$. Some of these frequency allocations are acceptable whereas others are not. For instance, the base stations 1 and 2 may be close to each other so that they should use different frequencies. As another example, base station 1 may be close to an unknown transmitter that uses frequencies 1 and 2. In that situation, base station 1 should use frequencies other than 1 and 2.

To cover all these various situations, we use the following definition of acceptable frequency allocations.

Definition 1 (Acceptable Frequency Allocations) *There is a nonempty subset $\mathbf{A}$ of $\{1, \dots, F\}^N$ that consists of acceptable frequency allocations. That set has the property that if $x \in \mathbf{A}^c$, then there is a subset $\mathbf{J}(x)$ of $\{1, \dots, N\}$ such that $y \in \mathbf{A}^c$ whenever $y_j = x_j$ for all $j \in \mathbf{J}(x)$. Moreover, all the base stations in $\mathbf{J}(x)$ detect that the frequency allocation is*

unacceptable.

This definition means that if a frequency allocation is unacceptable, i.e., belongs to $\mathbf{A}^c$, the complement of $\mathbf{A}$, then there is a subset $\mathbf{J}(x)$ of the base stations whose frequencies are not acceptable. When that is the case, the frequency allocation cannot be made acceptable by modifying only the frequencies of the base stations not in $\mathbf{J}(x)$. The definition covers the case of direct interference with unknown transmitters or between base stations.

For instance, consider a graph with N vertices. The vertices represent the N base stations. A branch between two vertices indicates that the corresponding base stations are geographically close to each other and that they should use different frequencies. In this situation, a frequency allocation is acceptable if all vertices connected by a branch are allocated different frequencies. The set $\mathbf{A}$ of acceptable frequencies satisfies the property of Definition 1. Note that in this case an acceptable allocation of frequencies corresponds to a valid coloring of the graph with F colors.

The property of Definition 1 is the only one that we need to prove the convergence of our algorithm.

In the next section we introduce an algorithm and we show that it does not converge to an acceptable frequency allocation. This observation leads us to define another algorithm whose convergence is the subject of Section 4.

3 Frequency Allocation Algorithms

Our objective is to derive an asynchronized distributed algorithm that converges to an acceptable frequency allocation. Such an algorithm does not require any communication among the base stations or between a centralized agent and the base stations.

We first define an algorithm that seems reasonable and we show, with an example, that it does not do the job.

Definition 2 (Algorithm 1) *Let $x \in \{1, \dots, F\}^N$ be some frequency allocation. If $x \in \mathbf{A}$, then the algorithm stops. If $x \in \mathbf{A}^c$, then every base station in $\mathbf{J}(x)$ has a positive probability of making the next move. Say that base station $j \in \mathbf{J}(x)$ makes the next move. That station attempts to find some frequency y_j in $\{1, \dots, F\} \setminus \{x_j\}$ such that j is not in $\mathbf{J}(x_1, x_2, \dots, x_{j-1}, y_j, x_{j+1}, \dots, x_N)$. If base station does not find such a frequency y_j , then it keeps the frequency x_j and the procedure continues with another base station. If base station j finds such a frequency y_j , then it uses it and the procedure continues.*

This algorithm attempts to eliminate the conflicts. The assumption that the conflicting stations (in $\mathbf{J}(x)$) all have a positive probability of making the next move reflects the distributed and asynchronous aspects of the algorithm. Note that the algorithm assumes that the station that makes the next move gets to complete that move before another station makes its own move.

Unfortunately, as we show next, the above algorithm may not have the desirable behavior.

Fact 1 (Algorithm 1 may not converge) *Algorithm 1 may fail to converge to an acceptable frequency allocation.*

To show this fact, we exhibit a simple example where Algorithm 1 fails to converge. In fact, in that example, Algorithm 1 gets stuck at an unacceptable frequency allocation even though there is an acceptable one.

Consider four base stations located at the corners of a square. The base stations can use one of two frequencies 1 and 2. Two base stations on a same side of the square cannot use the same frequency. Numbering the corners clockwise as 1, 2, 3, 4, we see that $\mathbf{A} = \{(1, 2, 1, 2), (2, 1, 2, 1)\}$. Assume that the initial allocation is $x = (1, 1, 2, 2)$. Then the algorithm is stuck in that unacceptable allocation. Indeed, no base station can find a new nonconflicting frequency. Note that $\mathbf{J}(x) = \{1, 2, 3, 4\}$. If base station 1 is the one that makes the first move, then it cannot choose the frequency 2 since $1 \in \mathbf{J}(2, 1, 2, 2)$. The situation is analogous for the other base stations.

A simple modification of Algorithm 1 turns out to converge. We define it next.

Definition 3 (Algorithm 2) *Let $x \in \{1, \dots, F\}^N$ be some frequency allocation. If $x \in \mathbf{A}$, then the algorithm stops. If $x \in \mathbf{A}^c$, then every base station in $\mathbf{J}(x)$ has a positive probability of making the next move. Say that base station $j \in \mathbf{J}(x)$ makes the next move. That station chooses some frequency y_j in $\{1, \dots, F\} \setminus \{x_j\}$ in such a way that all the frequencies in that set have a positive probability of being selected. The procedure continues.*

This algorithm appears less efficient than Algorithm 1 since it may not eliminate a conflict at each step. In fact, the algorithm may create more conflicts.

Let us revisit our example and examine the steps of Algorithm 2. We start as before with $x = x^0 := (1, 1, 2, 2)$. Say that base station 1 makes the first move. The allocation becomes $x^1 := (2, 1, 2, 2)$. If station 4 makes the next move, then the allocation becomes acceptable. If station 1 or 3 makes the next move, then the allocation becomes $(1, 1, 2, 2)$ or $(2, 1, 1, 2)$ and we are back to square one. If station 2 makes the next move, the allocation becomes $x^2 := (2, 2, 2, 2)$. The next step is then $(1, 2, 2, 2)$ or a cyclic permutation. Note that this state is similar to state x^1 . From x^0 the next state could have been $y^1 := (1, 1, 1, 2)$, which is similar to x^1 with the frequencies 1 and 2 interchanged. This discussion shows that the transition diagram is as indicated in Figure 1 where we do not distinguish cyclic permutations. The diagram shows that the successive states form a Markov chain with one absorbing state. Thus, the algorithm will eventually reach that absorbing state which is an acceptable frequency allocation.

In the next section, we show that Algorithm 2 converges under the general assumptions of Definition 1.

Figure 1: Transition diagram for 2 frequencies and 4 base stations located at the corners of a square.

4 Proof of Convergence

In this section we prove the following result.

Theorem 1 (Convergence) *Algorithm 2 always converges to some element in $\mathbf{A}$.*

Proof:

Let $z \in \mathbf{A}$ and assume that the current allocation x is not in $\mathbf{A}$.

We claim that there is some

$$j \in \mathbf{J}(x) \text{ such that } x_j \neq z_j.$$

To see why this claim is valid, first observe that the set $\mathbf{J}(x)$ is not empty since x is not in $\mathbf{A}$. Next note that Definition 1 implies that

$$\text{if } x_j = z_j \text{ for all } j \in \mathbf{J}(x), \text{ then } z \in \mathbf{A}^c. \quad (1)$$

Indeed, x cannot be transformed into the acceptable z by modifying only the frequencies at the base stations not in $\mathbf{J}(x)$. Thus Equation (1) proves the claim by contradiction.

According to the algorithm, there is a positive probability that x will be transformed next by replacing x_j by z_j . If this new state is in $\mathbf{A}$, we have shown that there is a path from the initial state to a state in $\mathbf{A}$. If not, we can repeat the argument and argue that there is a positive probability that another base station k with $x_k \neq z_k$ will change its frequency to z_k . Continuing in this way shows that there is a sequence of states from any state x in $\mathbf{A}^c$ to some state in $\mathbf{A}$. Thus the sequence of successive frequency allocations forms a finite

Markov chain that eventually reaches an absorbing state in $\mathbf{A}$. □

How long does the algorithm take to converge? We explore that question in the next section.

5 Time to Convergence

We have proposed an algorithm and we have shown that it converges. From an implementation viewpoint it is desirable to know whether the algorithm converges fast enough to justify the quasistatic assumption and also for the fraction of time that the stations spend to allocate frequencies to be negligible. In this section we present a rather crude estimate on the time to convergence.

Definition 4 (Average convergence time) *For $x \in \{1, \dots, N\}$ let $\alpha(x)$ denote the average number of steps of the algorithm until it finds an acceptable frequency allocation.*

We will derive a crude upper bound on $\alpha(x)$. To derive this upper bound we argue as follows.

Let z be an acceptable allocation. From any other allocation x there is some probability at least equal to p that the next step of the algorithm decreases the number of components of x that differ from those of z . We will estimate p later. With probability $q = 1 - p$, in the worst case, the next step of the algorithm will increase by one the number of components of x that differs from those of z . Consequently, the number $k(t)$ of components from the allocation derived by the algorithm at step t that differ from those of z is dominated stochastically by a birth-and-death process with death rate p and with birth rate $1 - p$. That Markov chain is reflected at N : no birth is possible from state N . Assuming that this number of different components is equal to k for the initial state x , we conclude that $\alpha(x)$ is less than or equal to the expected hitting time $\beta(k)$ of 0 by the birth-and-death started in state k .

To calculate $\beta(k)$ we use the first step equations satisfied by the hitting times of Markov chains:

$$\beta(k) = 1 + p\beta(k-1) + (1-p)\beta(k+1), \text{ for } k = 1, \dots, N$$

where $\beta(N+1) := \beta(N)$ and $\beta(0) := 0$. Solving these equations assuming $p < 0.5$ we find

$$\beta(k) = A - A\rho^{-k} - ck, \text{ for } k = 1, \dots, N \tag{2}$$

where $A = \frac{(1-p)^{N+1}}{p^N(1-2p)^2}$, $c = \frac{1}{1-2p}$, and $\rho = \frac{q}{p}$.

We now need to estimate p , the probability that the next step of the algorithm reduces the number of components that differ from those of z . This event occurs if the base station, say

j , that makes the next move is such that $x_j \neq z_j$ and if, in addition, the frequency selected by j is z_j . To estimate the probability of that event, we assume that the algorithm is such that all the base stations are equally likely to make the next move and that all the $F - 1$ frequencies are equally likely to be selected. In the worst case, only one base station has a frequency that differs from z , hence $p = \frac{1}{N(F-1)}$.

Combining the above results we find that

$$\alpha(x) \leq \beta(k)$$

where $\beta(k)$ is given by Equation (2) with $p = \frac{1}{N(F-1)}$.

Numerical values are as follows. We take the worst case $k = N$ and we denote the upper bound $\beta(N)$ by $T(N, F)$ to reflect the dependency on F . We find

$$T(4, 2) = 232, \quad T(6, 3) \approx 2 \times 10^6, \quad T(20, 6) \approx 8 \times 10^{39}. \quad (3)$$

If $F \geq 3$, and $p \ll 1$, we can approximate $T(N, F)$ as follows:

$$T(N, F) \approx N^N (F - 1)^{N-1} (F - 2).$$

These estimates are somewhat disappointing but we should keep in mind that they are probably overly pessimistic.

They can be improved slightly by considering a birth-and-death model with non-constant birth and death rates. The probability of selecting an $x_j \neq z_j$ is proportional to the number $k(t)$ of positions in which x and z differ, or $d(k) = \frac{k}{N(F-1)}$. On the other hand, the birth rate will increase with $N - k$, yielding $b(k) = \frac{N-k}{N}$. We can find $\beta(k)$ from the following difference equation:

$$\beta(k) = 1 + d(k)\beta(k-1) + b(k)\beta(k+1) + (1 - b(k) - d(k))\beta(k), \text{ for } k = 1, \dots, N$$

where $\beta(N+1) := \beta(N)$ and $\beta(0) := 0$.

The solution is given by

$$\beta(k) = \sum_{j=0}^{k-1} \frac{F^N - \sum_{i=0}^j (F-1)^i \binom{N}{i}}{(F-1)^j \binom{N-1}{j}}. \quad (4)$$

Again, the upper bound can be found by putting $k = N$:

$$T(4, 2) = 21.3, \quad T(6, 3) = 82.7, \quad T(20, 6) \approx 3.7 \times 10^{15}.$$

The asymptotic behavior of the upper bound is given by

$$T(N, F) = \beta(N) < \frac{F-1}{F-2} \cdot F^N \approx F^N,$$

which is about the size of the state space. Note also that $\beta(N) \approx \beta(1) = F^N - 1$.

This upper bound will not be very tight for most instances for the following reasons:

1. It assumes that there is only a single solution. In practice, this will only be valid for the hardest problems. Let us assume we have a configuration with N base stations and F frequencies, allowing only one solution. It is clear that increasing F will result in more solutions, and hence a faster convergence. However, this is not reflected in the upper bound.
2. It does not take into account any information on the geometry of the problem: interference will only occur between adjacent base stations.

We can modify the model to take into account the fact that there are many solutions by increasing the death rate $d(k)$ with a constant factor α , with $1 \leq \alpha \leq F-1$. Similarly, the birth rate will be modified with a factor $\frac{F-\alpha}{F-1}$. If $\alpha = 1$, we obtain the original expression, while $\alpha = F-1$ reflects the other extreme: all frequencies except for one are good. We obtain then the following expression:

$$\beta(k) = \frac{F-1}{F-\alpha} \sum_{j=0}^{k-1} \frac{\left(\frac{F}{\alpha}\right)^N - \sum_{i=0}^j \left(\frac{F-\alpha}{\alpha}\right)^i \binom{N}{i}}{\left(\frac{F-\alpha}{\alpha}\right)^j \binom{N-1}{j}}. \quad (5)$$

This leads to an improved upper bound:

$$\begin{aligned} T(F, N) &\leq \frac{F-1}{F-2\alpha} \left(\frac{F}{\alpha}\right)^N \quad \text{for } 1 \leq \alpha < \frac{F}{2} \\ &\leq \frac{2(F-1)}{F} N 2^N \quad \text{for } \alpha = \frac{F}{2}. \end{aligned}$$

For $\frac{F}{2} < \alpha \leq F-1$, Equation (5) yields a smaller upper bound, but finding a good asymptotic expression is difficult. It is clear that the bound cannot be worse than the bound for $\alpha = F/2$.

The second problem can be solved at least partially by introducing a more accurate expression for the number of births. If there are k base stations with the wrong frequency, only $r \cdot k$ base stations will notice this and modify their frequency. This means that

$$b(k) = \min \left(\frac{N-k}{N}, \frac{r \cdot k}{N} \right). \quad (6)$$

The value of r can be determined from the geometry of the problem: if the base stations lie on a line, $r = 2$, while if they are on a regular hexagonal grid $r \approx 3$ (note that in the

simplest case a frequency assignment with 3 frequencies is feasible). The resulting difference equation can be solved numerically.

The upper bounds for a hexagonal configuration with 20 and 50 base stations are indicated in Table 1.

Table 1: Upper bounds for the hitting time for $N = 20$ and $N = 50$ base stations on a hexagonal grid. The first column combines all improvements on the bounds, including Equation (6) with $r = 3$.

# frequencies F	combined	Equation (5)	Equation (4)	Equation (2)
$F = 3, \alpha = 1$	$8.7 \cdot 10^7$	$3.6 \cdot 10^9$	$3.6 \cdot 10^9$	$7.2 \cdot 10^{31}$
$F = 4, \alpha = 2$	$5.0 \cdot 10^4$	$1.7 \cdot 10^6$	$1.1 \cdot 10^{12}$	$2.7 \cdot 10^{35}$
$F = 5, \alpha = 3$	$2.4 \cdot 10^3$	$6.0 \cdot 10^4$	$9.7 \cdot 10^{13}$	$9.3 \cdot 10^{37}$
$F = 6, \alpha = 4$	$5.9 \cdot 10^2$	$9.5 \cdot 10^3$	$3.7 \cdot 10^{15}$	$8.4 \cdot 10^{39}$
$F = 7, \alpha = 5$	$2.9 \cdot 10^2$	$3.0 \cdot 10^3$	$8.1 \cdot 10^{16}$	$3.3 \cdot 10^{41}$

# frequencies F	combined	Equation (5)	Equation (4)	Equation (2)
$F = 3, \alpha = 1$	$5.0 \cdot 10^{18}$	$7.3 \cdot 10^{19}$	$7.3 \cdot 10^{23}$	$6.2 \cdot 10^{99}$
$F = 4, \alpha = 2$	$1.5 \cdot 10^{10}$	$1.7 \cdot 10^{15}$	$1.3 \cdot 10^{30}$	$4.7 \cdot 10^{108}$
$F = 5, \alpha = 3$	$2.9 \cdot 10^6$	$2.6 \cdot 10^{11}$	$8.9 \cdot 10^{34}$	$8.9 \cdot 10^{114}$
$F = 6, \alpha = 4$	$3.0 \cdot 10^4$	$1.7 \cdot 10^9$	$8.1 \cdot 10^{38}$	$6.5 \cdot 10^{119}$
$F = 7, \alpha = 5$	$2.9 \cdot 10^3$	$6.4 \cdot 10^7$	$1.8 \cdot 10^{42}$	$6.1 \cdot 10^{123}$

The conclusion of this section is that the upper bound on the time to convergence is in general exponential in the number of base stations and in the number of frequencies. If however the number of frequencies is increased, convergence times which are subexponential in N can be expected.

6 Speeding up Convergence

In the previous section we have developed upper bounds for the time to convergence. In order to accelerate the convergence, we can optimize Algorithm 2. We will consider the following optimizations: the choice of the new frequency, imposing a fixed order on the base stations, the parallelization of the algorithm, and the choice of the initial configuration. These optimizations are partially inspired by related results in graph coloring [6, 8, 9].

6.1 Choosing a New Frequency

Algorithm 2 specifies that a new frequency is selected such that every frequency different from the original one has a positive probability. The first observation is that the algorithm will still converge when the original frequency is selected with a positive probability < 1 . This is also intuitively acceptable: let us assume that a node measures interference which is higher than the threshold. That node will then measure the interference at other frequencies, and it might well be that the interference at some of these frequencies is higher than at its current frequency. There is then no reason to select any of these frequencies with a higher probability than the original frequency.

Let us denote the interference measured by base station i in frequency f with $I_i(f)$. The probability of selecting frequency f is given by

$$\Pr(F = f) = \frac{\gamma^{-I_i(f)}}{\sum_f \gamma^{-I_i(f)}}. \quad (7)$$

The optimal choice for γ depends on the geometry of the problem. The choice of the exponential is based on the analogy with the energy function in statistical mechanics. Empirical evidence suggests that $\gamma = 4$ is optimal for 3-coloring graphs [6]. This value of γ was also used in [9]. Note that in [4], it was shown that this type of updating algorithm is natural in a Dynamic Channel Algorithm environment.

6.2 Imposing a Fixed Order on the Base Stations

One might consider to impose a fixed order on the base stations. This will guarantee that after N steps every base station has been activated. We can show however that this may lead to loops if a station is forced to choose a new frequency. A simple counterexample can be given for the case of the four base stations located at the corners of a square. If we impose the fixed order $1 - 2 - 4 - 3$, 4 initial states (namely $(1, 2, 1, 1)$, $(2, 1, 2, 2)$, $(1, 1, 1, 2)$, and $(2, 2, 2, 1)$) lead to a cycle with period 8. Note that these two cycles visit all states except for the 2 acceptable states and the states $(1, 1, 1, 1)$ and $(2, 2, 2, 2)$, in which no arrival is possible.

6.3 Parallelization of Algorithm 2

The main difficulty in designing an algorithm for our problem is caused by the requirement that there is no communication between the base stations. If all information would be centralized, a modification of heuristic algorithms for graph coloring would result in very efficient solutions. On the other hand, the fact that the algorithm is distributed with only local information processing can be used to parallelize it, which will result in a speed-up of the convergence roughly proportional to the number of base stations which are active simultaneously.

First we will show that parallelization of the algorithm can result in problems if the updates are synchronized and if the new frequency has to be different from the old one. The first

counterexample is for a configuration with 2 frequencies where the base stations lie on a line or on a circle. It is easy to see that if we start in a state where all frequencies are equal, the state will oscillate between the two states with equal frequencies. A similar situation occurs for a triangular graph with 3 frequencies. In that case the states with 1 or 2 different frequencies will form a cycle. It seems that if there are more frequencies, no such cycles exist. If the probability of staying at the same frequency is positive, the cycles are eliminated, but the convergence will still be slow. There is however also a practical problem: in order to measure the interference, a base station has to switch off its own transmitter. From this it follows that not too many base stations should update their frequencies simultaneously.

If we parallelize Algorithm 2, we have to prove again that the algorithm will converge. This will require that there is a positive probability that a base station keeps its frequency, even if it measures an interference level which is not acceptable. First we will assume that interference measurements do not require that the transmitter of that base station is switched off.

Theorem 2 (Parallel convergence I) *The asynchronous parallel version of Algorithm 2 always converges to some element in $\mathbf{A}$.*

Proof:

The proof of Theorem 1 can be extended to this case. Assume $M < N$ base stations will update their frequencies in parallel.

Let $z \in \mathbf{A}$ and assume that the current allocation x is not in $\mathbf{A}$.

The argument in the proof of Theorem 1 can be repeated to show that there is some

$$j \in \mathbf{J}(x) \text{ such that } x_j \neq z_j.$$

There is a positive probability that base station j will be one of the M base stations to update its frequency, and that x will be transformed next by replacing x_j by z_j . This is not sufficient to prove convergence: it might be that one or more of the other base station which had a correct frequency, will change their frequency again in this iteration. However, there is a positive probability that those base stations which had the correct frequency before this iteration and which have been activated, will not modify their frequency. $\square$

Note that this proof does not preclude that in one iteration more than one base station will converge to the correct frequency assignment.

Practical experiments for the case of graph coloring suggest that the speed-up is almost linear in the number of base stations. However, if too many base stations are active at once, the probability that base stations will choose new conflicting frequencies will become too large, and the algorithm will slow down. Therefore it is recommended that about 60% of the base stations should be active simultaneously [9].

We now study the case where the transmitter of a base station is switched off during interference measurements. Hereto we have to make some additional assumptions about the model. Let us assume that the interference in a single base station i will only be unacceptable if at least t_i base stations in its neighborhood are active in the same frequency. Define now $t = \max_i \{t_i\}$. It is clear that convergence requires that $M \leq N - t$: if the transmitters of $N - t + 1$ base station are switched off, only $t - 1$ base stations will transmit. This means that there will be at least one base station which will always measure less interference than the threshold, while its frequency is not acceptable. In practice one will choose a smaller value of M in order to achieve a faster convergence.

Theorem 3 (Parallel convergence II) *Let M and t be defined as above. If $M \leq N - t$, the asynchronous parallel version of Algorithm 2 where the transmitters of the base stations are switched off during measurements always converges to some element in $\mathbf{A}$.*

Proof:

With the condition on the parameters, the proof is essentially the same as the one of Theorem 2. One only has to note that if the frequency of a node is not acceptable, there is a positive probability that the M active base stations during the next iteration will not include the base stations which create the interference at that node. $\square$

Note that in the algorithm of [1] the interference measurements require that the base stations switch off their transmitters. Although in [1] the base stations act autonomously, the system parameters are set up such that it is very unlikely that two base stations will collide with each other, i.e., measure at the same time. In our algorithm we have removed this restriction.

6.4 Initial Configuration

Algorithm 2 assumes that initially the base stations pick their frequencies uniformly and independently from the set $\{1, \dots, F\}$. One could try to decrease the time to convergence by starting from a better initial configuration. However, since the algorithm does not use any direct communication between the base stations, it cannot always take advantage of an initial configuration which is close to an acceptable state. E.g., if there is a conflict which can be resolved by one base station, there is a positive probability that a neighboring base station will become active first, sense the conflict, and modify its frequency. In this way the configuration can move away from a nearby acceptable state.

7 The One-Dimensional Case

In this section we study a very simple configuration, where the base stations are located on a line, and experience only interference from their two direct neighbors. It is easy to see

that in this case two frequencies are sufficient. Deriving analytic expressions for the time to convergence becomes straightforward if more than two frequencies are available. It is then always possible for a base station which detects a conflict to resolve it by choosing a new frequency. The fastest convergence is obtained by choosing the constant γ of Equation (7) equal to $+\infty$. The results provide some useful insight towards more complicated instances of the problem.

In a first step we determine the number of initial conflicts, assuming that the initial frequencies are selected independently and uniformly. We first consider the $N-2$ base stations which are not at either end. The probability of a conflict at the left side is equal to $1/F$, which is also the probability for a conflict at the right side. By independence, the probability for a conflict both at the left and at the right side is equal to $1/F^2$. The base stations at the end points each have a probability of $1/F$ of detecting a conflict. The expected number C of conflicts is thus equal to

$$C = (N-2) \cdot \frac{2F-1}{F^2} + \frac{2}{F}. \quad (8)$$

Next we derive an expression for the number U of updates that are required to resolve all conflicts. The number of updates is upper bounded by the number C of conflicts, but it will in general be smaller, since a single update can resolve 1, 2, or 3 conflicts. If k is the maximal number such that k adjacent base stations detect a conflict, we say that the conflict chain has *length* k . In the following we will make abstraction from the effects of the end points: for large values of N these effects will be negligible. The expected number of conflict chains of length k is equal to

$$N \left(\frac{F-1}{F} \right)^2 \cdot \left(\frac{1}{F} \right)^{k-1}. \quad (9)$$

Let us denote the expected number of steps to resolve a conflict chain of length k with a_k . It is easy to see that $a_2 = 2$. The sequence a_k obeys the following recursion:

$$\begin{aligned} a_k &= \frac{2}{k}(1 + a_{n-1}) + \frac{2}{k}(1 + a_{n-2}) + \frac{2}{k}(a_2 + a_{n-3}) + \cdots + \frac{2}{k}(1 + a_{n/2-1} + a_{n/2}) & n \text{ even} \\ &= \frac{2}{k}(1 + a_{n-1}) + \frac{2}{k}(1 + a_{n-2}) + \frac{2}{k}(a_2 + a_{n-3}) + \cdots + \frac{1}{k}(1 + 2a_{(n-1)/2}) & n \text{ odd.} \end{aligned}$$

The solution is given by

$$a_k = \frac{2k-1}{3}. \quad (10)$$

Combining Equation (9) and (10) yields

$$U = \sum_{k=2}^N \frac{2k-1}{3} N \left(\frac{F-1}{F} \right)^2 \cdot \left(\frac{1}{F} \right)^{k-1}$$

If N is large this can be simplified to

$$U = \frac{N}{3} \frac{3F-1}{F^2}. \quad (11)$$

Finally we are ready to determine the number I of iterations. This number will be larger than the number of updates, since an update will only be made if the active base station detects a conflict. We will assume that the number of conflicts starts with C and decreases linearly in U steps towards 0. This is only an approximation: the number of conflicts will decrease faster in the beginning, since the splitting of larger chains of conflicts is more likely to occur in the first phase. The expected number of conflicts as a function of time can be approximated by

$$C(t) \approx N \frac{2F-1}{F^2} \left(1 - \frac{t}{U}\right) \quad \text{for } t = 0, 1, \dots, \lfloor U \rfloor - 1.$$

The expected number of iterations is then given by

$$I = \sum_{t=0}^{\lfloor U \rfloor - 1} \frac{N}{C(t)}.$$

In order to obtain a closed expression, we approximate the summation by an integral:

$$I = N \frac{3F-1}{3(2F-1)} \left[\ln \left(N \frac{3F-1}{3F^2} \right) + \gamma \right], \quad (12)$$

where $\gamma = 0.577216\dots$ is Euler's constant. Figure 2 shows the number of iterations for 3, 4, and 5 frequencies. We have verified the validity of the approximations by computer simulations.

Figure 2: Number of iterations for a linear configuration.

8 Simulation Results

First we present some results for a graph coloring model. Subsequently we use a more realistic model for the interference.

8.1 Simplified Model: Graph Coloring

In this section we give some results for the special case of graph coloring. This corresponds to an interference equal to 1 for adjacent base stations, and 0 for the other ones. Simulation experiments for 3-colorable and more general graphs have already been described in [6, 8, 9]. Our results verify how the algorithm performs on two special classes of graphs, and can be used for comparison with a more realistic propagation model.

We distinguish between two types of configurations. In a grid graph, the vertices lie on a regular grid, and every inner vertex has exactly four neighbors. In a hexagonal graph every inner vertex has exactly six neighbors. For each of these we put the graph on a torus, which avoids special effects near the boundary. We will refer to this type of graph as a *periodic* graph. The minimum number of frequencies equals 2 for a grid graph, and 3 for the hexagonal graph. Figure 3 depicts a hexagonal and a grid graph.

Figure 3: A hexagonal and a grid graph with 49 cells.

Figure 4 shows how the number of iterations grows with the number N of base stations; the number F of frequencies is a parameter. As expected, the number of operations will grow slower with the number of base stations if the number of frequencies is increased. A saturation occurs when there are more than 5 frequencies. Simulations show that the time to convergence for a non-periodic graph is only slightly smaller.

Two questions remain to be addressed: can the algorithm be parallelized efficiently, and

Figure 4: Number of iterations for a periodic grid configuration.

what is the optimal choice for γ ? The first question is answered positively for a different class of graphs in [8, 9], under the condition that not too many base stations are active at the same time. The second question is much more difficult to answer. It is easy to understand why γ should increase if the number of frequencies increases: e.g., if the number of frequencies exceeds the number of neighbors, every base station can resolve a conflict in one step. Experiments suggest that if the number of frequencies becomes very small, $\gamma \approx 2 \dots 3$ can lead to faster convergence. On the other hand, if more base stations are active in parallel, γ should be increased. In this case making the algorithm more conservative will speed up the convergence.

8.2 More Realistic Propagation Model

In this section we consider a more realistic propagation model and discuss the results of a large number of simulations.

Three elements effect the propagation of radio signals [7]: path loss, log-normally distributed shadow fading, and Rayleigh-distributed short-term envelope fading. The effect of envelope fading can be mitigated by using diversity selection. In [1] it is shown that for the simple quasistatic algorithm averaging out the Rayleigh process is not required. Also a model for shadow fading is proposed. We did not incorporate this in our simulations, since we believe that it will not affect the convergence properties of our algorithm. If it is necessary to predict the received signal power, our model can easily be extended to take this into account. The only attenuation in our model is thus path loss: the signal power decays according to $1/r^\beta$, where r is the distance from the base station and β is typically $2 \dots 4$,

where 2 is for the near field and 4 for the far field.

In our simulations we only consider cochannel interference. It is assumed that adjacent channel interference and thermal noise are sufficiently small.

We investigate the influence of the following parameters: the update parameter γ , the number P of parallel base stations, the interference threshold T , the propagation parameter β , the asymmetry of the configuration, and the number N of base stations.

8.2.1 The Update Parameter γ

A first problem is to determine the optimal value of γ . Since the interference is no longer an integer between 0 and the number of neighbors, and there is no straightforward way to normalize it, γ has to be made dependent on the interference level. We consider the following changes to the interference pattern:

1. all interferences are increased with a constant offset Δ ,
2. all interferences are multiplied by a constant factor Δ ,
3. the interference for only 1 frequency becomes very large.

Ideally, we would like that in the first two cases, $\Pr(F = f)$ will not change too much, and in the third case, the relative values of the other frequencies are maintained. Note that in practice the situation is more complex: if the power of all base stations is increased, the new interference is a function of the distance r and the value of β .

If we do not modify γ , the values of $\Pr(F = f)$ will not change in the first case, and will change only moderately in the second case. If the interference in one frequency increases, the relative size of the other probabilities remains unchanged.

If we normalize the interference, i.e., we divide $I_i(f)$ by $\sum_f I_i(f)$ before applying Equation (7), the values of $\Pr(F = f)$ will not change in the second case, and will change only moderately in the first case. However, if one of the $I_i(f)$ values becomes very large, the remaining probabilities tend to become equal, which is an undesirable property.

A third alternative is to scale according to the minimum of the $I_i(f)$. This results in the same behavior as the previous solution for the first and second case without equalizing the probabilities in the third case. The disadvantage of this approach is that a single very low interference will have a probability close to 1, even if the other interferences are below the threshold.

These compromises can be avoided by taking into account that the algorithm requires the selection of an interference threshold T to define an acceptable state. Therefore we will normalize $I_i(f)$ with respect to this threshold T or $\gamma = \exp(a/T)$, where a is a new parameter. We have chosen this exponential form since our algorithm will use a wide range of values for γ .

Note that in the simulations we normalize the interference by making the interference from the neighbors at unit distance in the grid equal to 1. This normalization is useful to compare different configurations, but is not required for the algorithm to work properly.

For a fixed interference and threshold, the choice of γ determines the nature of the convergence. For small values of γ , the convergence is very irregular, and if γ is too small, the convergence becomes very slow for all cases. For large values of γ , the convergence becomes smoother and faster. However, if γ becomes too large, the algorithm will become extremely slow for some cases.

8.2.2 The Number P of Parallel Base Stations

Parallelization is an important element in speeding up the algorithm. We have chosen for the following model: during a single iteration, $P < N$ base stations are simultaneously active. For each iteration the subset is chosen uniformly without replacements from the N base stations, and the choices for every iteration are independent. If no synchronization would be available between the base station, the iterations would be unslotted: every base station becomes active after a random waiting time. It is easy to see that this will not slow down the convergence of the parallel operation.

The speed-up is close to linear in the number of parallel base stations, if this number is at most 25% of the total number. The number of parallel iterations does not decrease if between 25 and 50% of the base stations are active. If more than half of the base stations become active, the number of parallel iterations will increase. The time to convergence depends only slightly on the value of a . In the following we will take $a = 2.0$ unless indicated otherwise. Note that if active base stations do not have to switch off their transmitter, the fastest convergence occurs when 50% of the base stations are active. The time to convergence is about three times smaller, but it depends strongly on the value of a .

Figure 5 shows the number of iterations as a function of the exponent a for different values of P , the number of parallel base stations. The configuration is a periodic grid graph with 64 (8×8) base stations, with $F = 7$ frequencies. The propagation parameter β is equal to 4 and the interference threshold T is equal to 0.22. There exists an optimal reuse configuration with $F = 4$ frequencies, which results in an interference level of 0.33 (1.8dB larger). If $P = 1$, the number of iterations is about 1000, and does not depend much on a as long as a lies between 0.5 and 2.5 (corresponding to $\gamma = 9.7058$ and 86 132 respectively). For $P \leq 16$ the speed-up is almost linear in P . For values of $P > 1$, it is better to take $a > 1$, and for $P > 48$ the convergence is faster if $a < 2.5$. It is also clear that choosing $P > 32$ does not offer any advantage.

8.2.3 The Interference Threshold T

The algorithm requires the definition of an acceptable interference level T . If a base station measures an interference at its frequency which is lower than T , it will not update its frequency in this step. The value of T is thus the maximum value of the interference in

Figure 5: Number of iterations as a function of the exponent a of the update parameter γ . The parameter P is the number of base stations which are active in parallel.

an acceptable configuration. One expects that if the threshold T decreases, the time to convergence will increase.

Figure 6 shows the number of iterations and the average interference as a function of the threshold T . For large values of T , the number of iterations decreases with decreasing T . The reason is that decreasing T forces more active base stations to update their frequency. For small values of T , it becomes harder to reach an acceptable configuration. When T approaches a critical value the number of iterations increases sharply. In an optimal configuration, the average and maximal interference would be almost equal: the time to reach such an optimum is however exponential in N .

Note that the expression $\gamma = \exp(a/T)$ seems to be counter-intuitive: if the threshold is decreased, finding a solution become more difficult, and γ should be decreased as well in order to speed up the convergence. It is however necessary to adapt γ to the interference level as explained in Section 8.2.1. For a fixed interference level, different thresholds might be used, and it is impossible to predict for which threshold the problem will be easy or difficult, especially when the interference becomes irregular. Fortunately the convergence speed does not depend too much on γ .

8.2.4 The Propagation Parameter β

We have repeated the simulations of the previous section for a grid configuration with $\beta = 3$. The behavior of the algorithm is identical, but the interference thresholds are larger. We

Figure 6: Number of iterations and average interference as a function of the interference threshold T . The parameter F is the number of frequencies.

have also simulated the convergence behavior for a hexagonal configuration with 7 or more frequencies, again with $\beta = 3$ and $\beta = 4$. Table 2 indicates the smallest threshold values for which the number of parallel iterations is smaller than 1000. For comparison, we have added the interference of the corresponding regular reuse configuration (marked with a *). Note that for a hexagonal graph, a periodic regular reuse graph requires at least 196 base stations. Therefore we have used in this case the non-periodic configuration as a reference; the interference in a periodic configuration is slightly larger. We can conclude that for these configurations, at most two additional frequencies are required to achieve a lower maximal interference than the regular reuse configuration.

Table 2: Interference threshold which can be reached in fewer than 1000 parallel iterations ($N = 64$, $P = 16$, $a = 2.0$). The first table is for a periodic grid configuration, and the second table is for a periodic hexagonal configuration.

F	$\beta = 4$	$\beta = 3$
4*	0.33	0.76
4	0.68	1.23
5	0.43	0.81
6	0.24	0.56
7	0.18	0.43
8	0.14	0.35
9	0.12	0.30

F	$\beta = 4$	$\beta = 3$
7*	0.13	0.36
7	0.18	0.44
8	0.14	0.36
9	0.12	0.30
10	0.13	0.32
11	0.11	0.27
12	0.072	0.22
13	0.061	0.18
14	0.055	0.16
15	0.051	0.15

8.2.5 Asymmetric Configuration

Because of practical constraints, the base stations will not be located on a perfect regular or hexagonal grid. Even if their locations would be perfectly symmetric, the different propagation conditions will destroy this symmetry. This is one of the reasons why allocating frequencies according to a regular pattern can cause problems during the deployment or extension of the system.

We use the following model: each the base stations is moved over a random distance, uniformly distributed in the interval $[0, e]$, in a random direction (the angle is uniformly distributed in the interval $[0, 2\pi[$). In [1] the behavior of a quasistatic algorithm is investigated when the base stations are placed randomly, under the restriction that their minimum distance is at least 0.2 times their average distance. For larger values of e , the base stations in our model can come arbitrarily close. Figure 7 illustrates some typical configurations which have been used: for $e \geq 1$ the locations appear randomly distributed, and several base stations are clustered.

The simulation results for this model are presented in Figure 8. The interference threshold

Figure 7: Coordinates of base stations for different values of e .

was chosen equal to the value which can be achieved with an optimal reuse configuration. If e increases slightly, the number of iterations decreases, since our algorithm can exploit the combination of the irregular structure with the additional frequencies. If e becomes larger than 0.6, the number of iterations will increase. It also is clear that nothing is gained from operating more than 25% of the base stations in parallel. An important observation is that for the regular configuration with $P = 1$ the value of $a \approx 1.2$ is optimal. However, with this value for a the number of iterations increases significantly if $e > 1$. Moreover, the variance of the distribution becomes very large, which is clearly unacceptable. The values of Figure 8 were obtained by decreasing a to 0.3 for $P = 1$. For larger values of P , this effect was not observed.

These observations can be explained as follows: the algorithm can sometimes get trapped in an unacceptable configuration, from which it is hard to escape. This problem can be resolved in two ways. Either we decrease a , which will increase the probability of accepting a conflicting frequency, or by activating P base stations in parallel. In the latter case, P base stations switch off their transmitter, which will decrease the interference in certain frequencies, which leads to the same result. This hypothesis was confirmed by performing simulations with 32 and 48 parallel base stations, where the active base stations did not switch off their transmitter. For $e > 1$ we observed again a large variance in the time to convergence, which could be reduced by reducing a .

Figure 8: Number of iterations for an irregular grid configuration as a function of the distance e . The parameter P is the number of base stations which are active in parallel.

8.2.6 The Number N of Base Stations

Since the algorithm is suited for parallelization, the number of operations will increase only slightly with the number of base stations. This is illustrated in Figure 9. The interference threshold is equal to 0.141, 0.331, 0.358, and 0.366 for $N = 16, 64, 144$, and 256 respectively, which corresponds to the interference level for the regular reuse configuration with $F = 4$. For $F = 5$ the time to convergence grows exponentially with N , but for $F \geq 6$ the increase is much smaller.

Figure 9: Number of iterations as a function of the number N of base stations. The parameter F is the number of frequencies.

8.2.7 Discussion of the Simulation Results

We have demonstrated that for a large number of configurations, our algorithm achieves an acceptable interference level after less than 100 parallel iterations. The algorithm is easy to implement, scalable, and works very well for irregular configurations, even if no minimal distance between the base stations is imposed. We also have explored the trade-off between the number F of frequencies and the interference threshold T . If the base stations are placed on a regular grid, and if the propagation conditions are perfectly homogeneous, our algorithm needs a few additional frequencies to achieve the same performance as a regular reuse configuration. The algorithm performs very well for an irregular configuration, even when base stations are collocated.

The only remaining parameter in the algorithm is the update parameter a . For the parallel configuration, where the active base stations switch off their transmitters, the time to

convergence does not depend very strongly on a . However, if only one base station is active or if the active base stations do not switch off their transmitters, the choice of a is more critical. For these cases we have developed an adaptive variant of the algorithm.

Figure 10 and 11 show the behavior of the maximal and average interference for $N = 64$ base stations on a grid with $e = 0$ and on an irregular grid with $e = 1.75$ respectively. The number of frequencies $F = 7$ and the update parameter $a = 2.0$. $P = 16$ base stations are active in parallel. The propagation parameter β equals 4. In Figure 11 the maximal interference becomes very large (the scale for the maximum is logarithmic), since some base stations are very close in this configuration (see Figure 7).

9 Concluding Remarks

We have proved the convergence of a simple quasistatic frequency allocation algorithm. We have studied several optimizations of the algorithm, including parallelization. If the interference requirements are not too strong, or if a sufficient number of frequencies is available, fast convergence can be expected. The algorithm is particularly useful if the interference pattern is irregular.

References

- [1] J.C.-I. Chuang (1991). Autonomous adaptive frequency assignment for TDMA portable radio systems. *IEEE Trans. Veh. Technol.*, **40**, 627-635.
- [2] L.J. Cimini, Jr., G.J. Foschini, and C.-L. I (1993). Distributed dynamic channel allocation algorithms for microcellular systems. in *Wireless communications. Future directions.*, J.M. Holtzmann and D.J. Goodman, Eds., Kluwer Academic Publishers, 219-242.
- [3] D. Duet, J.-F. Kiang, and D.R. Wolter (1993). An assessment of alternative wireless access technologies for PCS applications. *IEEE J. Select Areas Commun.*, **11**, 861-869.
- [4] R.J. McEliece, E.R. Rodemich, and S.K. Deora (1993). Why channel assignment algorithms are asymptotically smart. *Proceedings of the 1993 Allerton Conference*, 207-216.
- [5] A.J. Motley (1987). Advanced cordless telecommunications service. *IEEE J. Select Areas Commun.*, **5**, 774-782.
- [6] A.D. Petford and D.J.A. Welsh (1989). A randomized 3-coloring algorithm. *Discrete Mathematics*, **74**, 253-261.
- [7] M.D. Yacoub (1993). Foundations of mobile radio engineering. *CRC Press*.
- [8] J. Zerovnik (1990). A parallel variant of a heuristical algorithm for graph coloring. *Parallel Computing*, **13**, 95-100.

Figure 10: Evolution of maximal and average interference ($e = 0$).

Figure 11: Evolution of maximal and average interference ($e = 1.75$).

- [9] J. Zerovnik (1992). A parallel variant of a heuristical algorithm for graph coloring – Corrigendum. *Parallel Computing*, **18**, 897-900.