Performance Analysis of Channel Assignment Algorithms in Cellular Networks

YIHUI TANG

A THESIS
IN
THE DEPARTMENT
OF
COMPUTER SCIENCE

PRESENTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF MASTER
OF COMPUTER SCIENCE AT CONCORDIA
UNIVERSITY
MONTREAL, QUEBEC, CANADA

0-612-72946-X

ABSTRACT

Performance Analysis of Channel Assignment Algorithms in Cellular Networks

Yihui Tang

Efficient allocation of radio spectrum becomes increasingly important as the demand for wireless communication has been growing rapidly over the past two decades. Several channel assignment algorithms for increasing channel utilization have been studied. In Fixed Channel Assignment (FCA), channels are permanently allocated to each cell for its exclusive use, whereas in Dynamic Channel Assignment (DCA) the channels are put in a shared pool and dynamically assigned upon request. One example of a DCA scheme is the so-called *greedy strategy*, where all the channels are kept in a central ordered list and every node uses the first channels which are available. In Fixed Preference Assignment (FPA), each cell has a fixed preference list of all channels and the first available channel in the list will be used to serve an incoming call. In this thesis, we show upper bounds and lower bounds for the competitive ratios of the greedy and FPA strategies for different values of reuse distance. The average case performance analysis of these three schemes is also conducted using computer simulation.

Contents

List of Figures

List of Tables

Acknowledgments

I am deeply indebted to my advisor, Dr. Lata Narayanan, who has been a constant source of inspiration and encouragement through every stage in the preparation of this thesis. It has been extremely pleasant to work with her, and I am grateful for her guidance and generous financial support.

I would like to thank Dr. Jaroslav Opatrny and Dr. Hovhannes Harutyunyan for their time to read this thesis and giving me valuable comments.

Chapter 1

Introduction

In the past two decades, there has been a tremendous increase in the demand for wireless telephony and wireless data services. Cellular wireless communication has grown to be a significant part of the communication infrastructure. On the other hand, the radio frequency spectrum allocated for the cellular mobile system has been limited. As a result, the radio frequencies must be efficiently used to support the ever increasing demands. In this thesis, we study the problem of channel assignment: how to assign radio channels to calls in a cellular network, based on FDMA or TDMA technology.

In the following sections, we provide a brief background on cellular networks, and define the channel assignment problem in greater detail.

1.1 Cellular Concept

The tremendous growth of mobile communications could not have been achieved without the use of the *cellular* concept. Before that, the approach to mobile communication was quite similar to radio or television broadcasting: the coverage of an area was provided by setting up a single high-power transmitter on top of the highest point in the area and sending out the signal to the whole coverage area. The available radio spectrum was divided into a number of channels, each was dedicated to a specific user and all the users talked to the same transmitter. The number of users was limited by the number of available channels, which were locked up over the whole coverage area by a small number of calls. For example, a wireless phone service provider serving ten thousand customers would need ten thousand different channels to operate, although only a small fraction of them would be in actual use at any given time.

The number of channels required can be reduced by *temporal* and *spatial* reuse of radio channels. Temporal reuse, also called *trunking*, means using the same channels for different users at different times. An end-device will be assigned a channel only when it requests one for the call. Although trunking can use the radio spectrum resource more efficiently, the system capacity is still rather limited. The number of simultaneous calls is limited to the number of channels available. Since radio spectrum has historically been a scarce resource, this limits the system capacity quite severely. For example, the Bell mobile system in New York City in the 1970s, which used trunking, could support only 12 simultaneous callers. The other approach of using the radio channels more efficiently is spatial channel reuse. The users can use the same channel at the same time in non-adjacent geographical areas. The spatial reuse of channels is not possible in a centralized broadcasting network, but instead, the network is restructured in a distributed fashion.

Spatial channel reuse is one of the key concepts used by a cellular network, to achieve efficiency in using spectral resources. The other two main features of cellular networks are *cell splitting* to deal with increased demand, and *handoff* of calls moving from one cell to another. We describe each of these features in greater detail in the rest of this section.

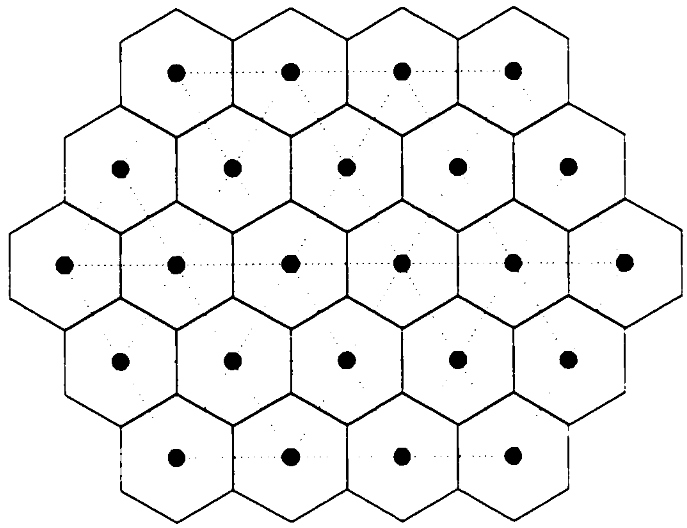

Figure 1.1: Hexagonal cellular network

1.1.1 Channel Reuse in Cellular Networks

To achieve a higher efficiency in channel usage through spatial channel reuse, the service area is divided into many adjacent regions. A *cell* is viewed as the approximate coverage area of a particular geographic region. Each cell has its own transmitter that is in wireless communication with mobile phones in its local area and is wired to a central office. The cellular concept described above was first introduced by MacDonald [21] using hexagonal cellular geometry to represent a cell as depicted in Figure 1.1. The reason for a hexagonal cell structure is that among all the geometric structures with the same radius that can cover a region without any gaps, a hexagon has the maximum area.

Unlike the traditional broadcasting-like approaches, the cellular idea deals with the coverage problem quite differently. Instead of covering a large area with just a single high-power transmitter, a cellular network provides the coverage of the area by using a lot of low-power transmitters, each specially designed to serve only a small area(cell) with radius as small as a few hundred meters. By dividing the coverage area into a large number of small cells each with its own transmitter, it becomes possible(at least in theory) to reuse the same channels across the area in different cells.

Small cells with channel reuse could increase traffic capacity substantially. To understand how this works, imagine that there are 12 channels available in a city, and that the city is covered by 100 cells. If all channels can be reused in every cell, then with the same number of 12 channels, instead of 12 simultaneous calls in the entire city, there would be 12 channels for every cell and there would be as many as 1,200 simultaneous calls in the city.

However, in reality, such extensive reuse is not possible. If the same channel is used in two different cells that are geographically close to each other, there can be radio interference that distorts the signals. This phenomenon, called *co-channel interference*, may lower the *signal to noise ratio (SNR)* to such a level that the signal is no longer distinguishable from the noise, when another user happens to be using the same channel in the next cell. In order to achieve an acceptable SNR, the same channel should not be reused in two different cells in the network, unless they are separated by a minimum distance [1], which is called the *reuse distance* σ.

Although the reuse distance constraint makes it necessary to skip one or several cells before reusing the same channel, the basic idea of channel reuse in the cellular concept remains valid. The same channel could be used to support more than one ongoing call in different parts of the city. This is possible because due to path loss of radio propagation, the average power received from a transmitter varies according to the inverse of the cube of the distance from the sender, or even a higher exponent up to the fifth or sixth power depending on the physical environment. As a result, if the inverse of the fourth power of distance is assumed, the SNR can be calculated as

[1] The distance between two cells is the minimum number of cell boundaries that must be crossed to go from one to the other.

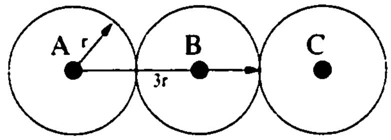

Figure 1.2: Channel reuse

below:

$$SNR = \frac{Signal\ Power}{Noise\ Power} = \frac{\alpha(1/d_S)^4}{\alpha(1/d_N)^4} = (\frac{d_N}{d_S})^4 \tag{1.1}$$

where $d_S(d_N)$ is the distance between the signal (noise) source and the user, and α is the physical constant of the environment. As we can see from Equation 1.1, SNR is determined not by the geographical distances d_S and d_N, but instead, the ratio of them. This makes it possible to use a graph theoretical representation of reuse distance constraints in cellular networks.

As shown in Fig 1.2, assume that every cell is of the same radius r. Then any user in cell A is at distance at most r away from its transmitter. The distance between cell A's transmitter and another user in cell C is at least $3r$. Therefore, if the power of the cell A's transmitter is at a level that is just adequate for everyone in cell A to hear the signal, the signal power received by any user in cell C will be $(\frac{1}{3})^4 \approx 1\%$ of that. The noise from the transmitter in cell A can hardly cause any distortion to communications in cell C. In modern systems, a reuse distance 2 or 3 may be enough to guarantee that the signal received from the main transmitter dominates the noise from the other transmitter using the same channel.

If reuse distance 2 is assumed, phones in adjacent cells are guaranteed to use different sets of channels. Non-adjacent cells may, however, use the same channel. For example, in Figure 1.2, cells A and B are adjacent, so they cannot use the same channel. However, calls in cells A and C may use the same channel.

The fact that the effects of interference are not related to absolute distance, but the ratio of the distance between cells to the radius of the cells makes the cellular idea even more attractive. The cell radius is determined by the transmitter power and by simply increasing or decreasing the transmitter's power level. the system operators can change the number of cells in the system and in turn. the number of calls that would be supported through reuse. For example. if a reuse distance of 3 is necessary for an acceptable signal to noise ratio and a grid of 10-mile-radius cells allowed reuse of the frequencies in a cell at a distance of 30 miles. then a grid of 5-mile-radius cells would allow reuse at 15 miles. and 1-mile-radius cells would allow reuse at 3 miles. Without adding more channels. a system based on 1-mile-radius cells would support one hundred times as many users as a system based on 10-mile-radius cells.

Of course. if we can indefinitely decrease the size of cells. the spectrum shortage problem can be easily solved by simply installing unlimited number of ultra small cells. However. the high cost of installation and maintenance and the increased control complexity make this an unfavorable solution. It is important to better utilize the resources in the existing system before changing to a smaller-cell system.

1.1.2 Cell Splitting

As the user population increases. there may no choice but to use more smaller cells to support increased demand in some areas such as downtown area of a city. But it would be too expensive to replace the entire cellular communication infrastructure with an all small-radius cell system. It appeared. however. that by using a technique called *cellsplitting*. large-radius cells could evolve gracefully into small-radius cells over a period of time. When the traffic in a cell has reached the point such that the existing allocation of channels in that cell is no longer able to support the increased traffic. new transmitters with lower transmitter power are installed and each covers a smaller area within the area of the former cell. By subdividing the cell into a number of smaller cells. the same channels assigned to the former cell could be reused within the original cell. Therefore, the number of supported users is effectively increased without disrupting any other cells in the system. This cell-splitting process can be repeated to support even more users as needed.

The flexibility in the dimensions of time and space has made cell-splitting an appealing technique for increasing the capacity as the system grows. The system can start with few cells and initial equipment investment could be very low. As the number of revenue-generating customers increases, new cells and equipment can be added. Moreover, the expenses of adding more smaller cells would only be necessary in the areas with high traffic density. Otherwise, few large cells would suffice to support the low traffic in the regions. Also, this expansion of the system can be done without wasting the existing investment, when a large cell is divided into more small cells, the transmitter of the formal cell will not be disbanded, but instead, it would fit within the new scale by simply being powered down.

However, cell splitting also has its drawbacks which limit its wide use in practice. The cost of setting up many small transmitters is high enough to make the network operator use the existing equipment more efficiently before adding more cells than is absolutely necessary. Besides the cost of more equipment, with more smaller cells in the network, the system becomes much more difficult to manage.

1.1.3 Handoff

The complexity of system control increases with a smaller cell system. With the cells' size scaled down to even a few hundred meters, it happens more and more frequently that a mobile call can not be completed within the boundaries of a single cell. A user in a moving car might cross several very small cells in a single conversation. Without a communication link promptly set up between the user and the transmitter in the new cell, the ongoing call would be annoyingly lost. To deal with this, a sophisticated *hand-off* technique is used. The movement of an ongoing call is continuously monitored through measurements of signal strength received from the mobile phones. The cellular system would be able to sense when a mobile with a call in progress was passing from one cell to another, and to switch the call from the current cell to the next cell without dropping or disrupting the call in progress.

1.2 Channel Assignment

Limited radio spectrum resources, and high cost and complexity of small cells motivated the study on efficient use of radio channels in cellular networks. Generally speaking, the *channel assignment problem* in a cellular network is the problem of assigning frequency channels to communication sessions, in such a way as to avoid interference. The objective is to use as few channels as possible to provide the maximum possible number of users an acceptable quality of service.

In this thesis, we take a graph theoretical approach to the problem of channel assignment. A cellular network is represented by a graph. Nodes in the graph represent the base stations and edges between cells represent interference constraints. At any instant of time, each node has a set of calls that it is providing channels to. We study both the offline or static version of the problem, where the set of calls remains fixed, and the online version of the problem where the set of calls to be served changes dynamically with time. The objective of the channel assignment algorithm is to assign channels to all calls, while respecting interference constraints, and to minimize the total number of channels used. We measure the performance of channel assignment algorithms using the standard yardsticks of performance ratio (for offline algorithms) and competitive ratio (for online algorithms). An offline channel assignment algorithm is said to have *performance ratio* k if it uses at most k times the minimum number of channels required. An online channel assignment algorithm has *competitive ratio* c if uses at most c times as many channels overall as the optimal offline algorithm would.

In another version of the problem, the number of channels available is limited. Therefore, all calls cannot be accepted. When a call is initiated but the channel assignment algorithm fails to assign it a channel, the call will be blocked. In this case, another performance metric can be used to evaluate the performance of the algorithm. Given certain number of channels, the objective of the channel assignment algorithm is to minimize the number of blocked calls. We also study other characteristics, such as fairness, capacity and channel usage of such algorithms.

1.3 Contribution of this Thesis

In this thesis, we propose and analyze algorithms for channel assignment in cellular networks. We study both the offline and online versions of the problem. Our main results are summarized below.

- For reuse distance 2, we show a tight bound of 5/3 on the performance ratio of the offline greedy algorithm. We also show a lower bound of 2.5 on the competitive ratio of the online greedy algorithm (Chapter 3).
- For reuse distance 3, we show that the performance ratio of the offline greedy algorithm lies between 7/3 and 23/8. We prove that the competitive ratio of the online greedy algorithm lies between 3 and 4 (Chapter 3).
- We show bounds on the performance ratio of the greedy algorithm as well as a borrowing strategy for any arbitrary reuse distance r (Chapter 3).
- We propose a new online version of the previously proposed FPA algorithm that does not perform reassignments and show a lower bound of 2.25 on its competitive ratio (Chapter 4).

- We also ran extensive computer simulations of online versions of FCA, greedy and FPA algorithms (Chapter 5). We studied the blocking probability, fairness index, capacity, and channel usage of these algorithms. Our results show that FCA has the lowest blocking probability when the traffic is light and even, while FPA performs the best over a wide range of traffic loads. The greedy algorithm's advantage over the two other algorithms is the fact that the system capacity remains relatively stable with variation in traffic loads, like the presence of hot spots. In terms of channel usage, FPA is shown to have more in common with FCA than DCA.

1.4 Thesis Outline

The remainder of this thesis is organized as follows: In Chapter 2, we present the formal definition, and a survey of previous work on channel assignment algorithms. In the next two chapters, we present the analytical results on the performance bound of two channel assignment algorithms: Dynamic Channel Assignment (Chapter 3) and Fixed Preference Assignment (Chapter 4). Chapter 5 contains the queuing models, simulation results and comparison between the algorithms mentioned above and fixed channel assignment, on aspects of blocking probability, system capacity, fairness, and local channel usage. We conclude in Chapter 6 with a brief summary of our work and some future directions for this research.

Chapter 2

Channel Assignment-A Literature Survey

The main challenge in the design of a wireless communication system is to meet the large user demand with limited radio spectrum resources. The primary technique used to increase the capacity of a cellular communication system is channel reuse. However, channel reuse is limited by the phenomenon of co-channel interference and the high cost associated with the smaller cell system. Effective channel assignment strategies are essential to the efficiency of spectrum reuse.

In this chapter, we first give an overview of the channel assignment problem in a cellular environment and then discuss three major categories of channel allocation schemes. Details of different algorithms are provided for each category.

2.1 The Channel Assignment Problem

At any given time in a cellular network, a certain number of active call connections are serviced by their nearest base station (transmitter). This service consists mainly of assigning a radio channel to each client call in a manner that the radio interference between two distinct calls in the network is below the acceptable level. The challenge is to find channel assignment strategies that exploit the principle of maximal channel reuse without violating the reuse constraints so that blocking is minimal. As discussed in Section 1.2, the signal-to-noise ratio is related only to the ratio of the distance to the signal source and the distance to the noise source. The co-channel interference constraint therefore can be adequately abstracted as the reuse distance constraint in a hexagon graph. This translation enables the graph theoretical approach to study the channel assignment problem.

Cellular communication networks are modeled as planar graphs [2] with each vertex representing a base station in a cell in the network and edges representing geographical adjacency of cells. In particular, cellular networks are usually modeled as *hexagon graphs*, which can be defined as finite induced sub-graphs of the infinite triangular lattice (see Figure 1.1).

Let G be a hexagon graph, and let the parameter r be the reuse distance. A *weighted graph* is a pair (G, w) where w is a positive integral vector indexed by the nodes of G. The component of w corresponding to node v is denoted $w(v)$ and is called the *weight* of v. The weight of a node represents the number of calls to be served at that node. The channel assignment problem with interference constraints can be modeled as a *multi-coloring problem* in graphs.

Given $G = (V, E)$, where $E \subseteq V \times V$, define $E^1 = E, E^k = \{(u, v) | \exists w \in V : (u, w) \in E^{k-1} \wedge (w, v) \in E\}$. The graph $G^r = (V, E')$ is defined by $E' = E \cup E^2 \cup \ldots \cup E^{r-1}$. Thus any pair of nodes at distance $i < r$ in G is connected by an edge in G^r. The problem of channel assignment in a weighted hexagon graph (G, w) with reuse distance r can be directly translated to the problem of multi-coloring the graph G^r. In particular, we are interested in a minimum multi-coloring of G^r that uses the least number of colors.

A multi-coloring provides a useful abstraction of the essential interference constraints: each color represents a distinct channel and it is assumed that two calls may use the same channel if and only if they originate in distinct cells that are not within the distance of r.

We define the *unweighted clique number* of G^r to be the maximum size of any clique in G^r, and is denoted $\omega(G^r)$. Similarly, the *chromatic number* of G^r, the minimum number of colors needed to color G^r, is denoted by $\chi(G^r)$. It is known [23, 32] that $\chi(G^r) = \omega(G^r)$, and an optimal coloring can be computed in polynomial time. We assume that such an optimal coloring of the graph G^r is available; thus every node in G^r is assigned a color from the set $\{1, 2, \ldots, \chi(G^r)\}$.

Given a weighted hexagon graph (G, w), we define $D_r(G, w)$ to be the maximum total weight on any clique in G^r. When solving the channel assignment problem on (G, w) with respect to reuse distance r, clearly $D_r(G, w)$ is a lower bound on the number of channels required. In the sequel, we sometimes use D_r when G and w are clear from the context.

So far, we have described a graph-theoretical model for the offline channel assignment problem. There are variations of this problem which have also been studied.

2.1.1 Offline vs. Online

Although the offline model serves as the foundation of the channel assignment problem, channel assignment is essentially an online problem [14]. At every time step, each node has associated with it a *set of calls*. In successive time steps, new calls may arrive, and old calls may be terminated. The algorithm must update the channel assignment accordingly. The online algorithm can also be implemented by running the offline algorithm, *i.e.* the algorithm on a single set of weights assigned to each node, at every step on the new weight vector [14]. Although this is a simple solution to the online problem, it may require changes to the existing assignment at every step and it is very uneconomical switching many ongoing calls to other channels especially in a wireless environment. On the other hand, generally speaking, at the cost of frequent reassignments, the offline approach can achieve more efficient channel reuse than an online algorithm in which the channel assigned to a call shall not be changed until the call is terminated.

We use standard yardsticks for measuring the efficiency of offline and online algorithms. An offline channel assignment algorithm is said to have *performance ratio* k if it uses at most k times the minimum number of channels required. An online channel assignment algorithm has *competitive ratio* c if uses at most c times as many channels overall as the optimal offline algorithm would.

2.1.2 Limited Number of Channels

Trying to satisfy all the calls using fewer channels may still result in using any number of channels. But in a real system, the number of channels is limited to some number C. This means not all calls can be necessarily accepted and a call will be blocked if it is unable to acquire a channel. As the blocking of calls is unavoidable with the limited number of channels, we are interested in algorithms that attempt to minimize the number of blocked calls.

In the offline version of this problem, the calls are assumed to have infinite duration and therefore, the call vectors are fixed. The objective of the algorithm is to find an assignment of the available channels to the call vectors and maximize the number of calls accepted using the limited number of channels.

There has been a tremendous amount of effort in studying the online version of this problem (also referred to as *admission control* problem [5]) using queuing theory and computer simulation. In the online version of the problem, the call vectors keep changing, which makes the admission control more difficult. The online version is of more importance because it reflects the reality that the calls come and go and their durations are not necessarily the same. When a new call arrives, it is the admission control algorithm's decision whether to accept the call and assign a channel to it, or to reject it. Once a call is accepted, the channel assigned to the call cannot be changed. The objective of admission control in a cellular network is to assign channels to calls in such a way that the blocking probability is minimized. In this thesis, we also study channel usage, capacity, and fairness of different algorithms. Details of these metrics are given in Chapter 5.

2.2 Fixed Channel Assignment

In *Fixed Channel Assignment (FCA)* [21], all the channels are divided into sets and each base station is assigned one set of channels. The base station can only use the channels from the fixed set assigned to it. The assignment is permanent and is precomputed to avoid interference with neighbors.

FCA defines a definite relationship between each channel and each cell, in accordance with co-channel constraints. The number of channel sets N required to serve the entire coverage area is decided by the reuse distance r as we need more channel sets with larger reuse distance value. Figure 2.1 gives the allocation of channel sets to cells for reuse distance two ($N = 3, r = 2$) and three ($N = 7, r = 3$), respectively.

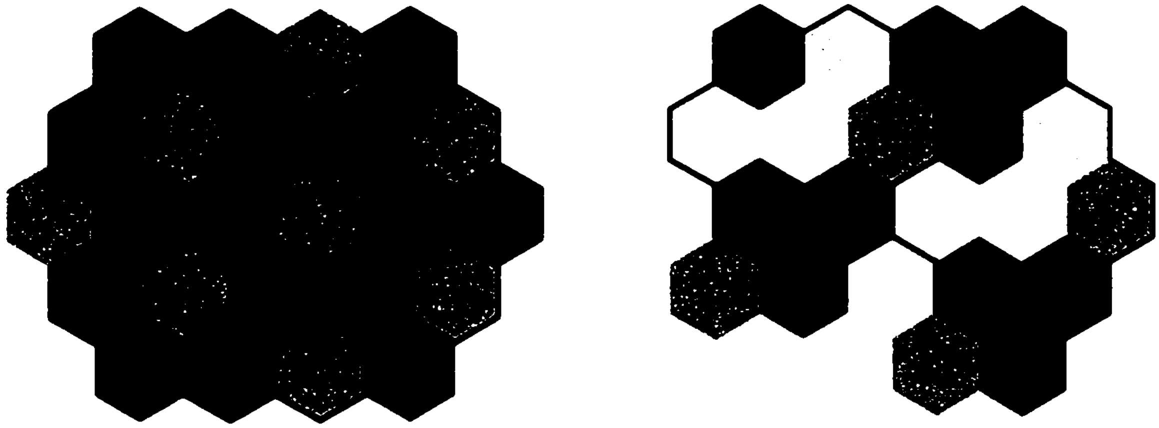

Figure 2.1: Reuse pattern $a)N = 3$; $b)N = 7$

In the simple FCA strategy, each cell is allocated the same number of channels. Without violating the constraint on the maximum clique weight, any node v could have a maximum weight of D_r where r is the reuse distance. In order to provide enough channels to every node even in such extreme cases, the FCA algorithm has to assign a set of D_r channels to every node in such a way that no two nodes in a clique will have the same set of channels. Therefore, the total number of channels FCA needs is equal to $\chi(G^r) \times D_r$ and this number is the same under both offline and online settings.

For example, as shown in Figure 2.1, the clique for reuse distance 2 consists of $\chi(G^2) = 3$ nodes and each node is assigned D_2 channels for its permanent use. Although every clique gets $3D_2$ channels, at any moment, there are at most D_2 channels are in use, therefore FCA has performance ratio 3. Similarly, when the reuse distance is 3, each clique has 7 nodes and therefore the performance ratio of FCA is 7.

A number of papers study the performance of online FCA for the situation when the number of channels available is limited. A call request in a cell will be blocked if all the channels allocated to that cell are being used. As we shall see in Chapter 5, with the FCA strategy, the probability of a call being blocked in a cell is proportional to the ratio of traffic to the number of channels in that cell. Therefore, the uniform channel distribution is efficient if the traffic distribution of the system is also uniform. In that case, the blocking probability in a cell is the same everywhere in the network. The overall average blocking probability of the system is the same as the call blocking probability in a cell. On the other hand, if traffic in a cellular system is non-uniform, a uniform allocation of channels to cells may result in high blocking in some cells with more traffic, while many channels are rarely used in those cells with less traffic. Intuitively, this poor channel utilization could be improved by allocating the channels in a cell to match the load in it. Heavily loaded cells, for example, cells in the downtown area, are assigned more channels than lightly loaded ones in the suburb area.

In [35], a nominal channel allocation algorithm called *non-uniform compact pattern allocation*, is proposed for allocating channels to cells according to the traffic distribution. The proposed technique attempts to allocate channels to the cells in such a way that the average blocking probability in the entire system is minimized. Let there be N cells and M channels in the cellular system. The allocation of a channel to a set of co-channel cells forms a pattern called *allocation pattern*. Without violating the channel reuse constraint, there are sizable number of possible allocation patterns for each channel. A heuristic algorithm is developed for searching the optimal allocation pattern to minimize the overall average blocking probability.

The M channels are allocated to the system one channel at a time. In order to reduce the amount of search effort, an optimal allocation pattern is assumed for every channel. The *compact allocation pattern* is defined as the allocation pattern with minimum average distance between co-channel cells. Given the traffic load in each cell in the system and all the possible compact allocation patterns for the channel, the allocation pattern is chosen to minimize the blocking probability without changing the allocation of the previous channels.

In [16], a novel way of using an FCA strategy is presented. The cellular network is partitioned into cliques or *clusters.* Instead of assigning sets of channels to individual cells, they are assigned to a *cluster of cells.* The algorithm is called *cluster partitioning.* Every cluster is assigned D_r channels for use, and nodes within the clique are assigned channels in an arbitrary manner. The interesting fact is that for any value of reuse distance, the cliques themselves can be 4-colored, so that any two nodes in cliques that have the same color are at distance at least r from each other. Thus $4D_r$ channels suffice to complete the assignment; leading to a competitive ratio of 4 for this algorithm. In fact, for reuse distance 2, the cliques can be 3-colored, which implies a competitive ratio of 3.

In this thesis, we run simulations of FCA to compare with other algorithms. No prior knowledge of traffic distribution is assumed and the channels are divided into a number of equal size sets. We use the standard version of FCA where all cells have the same number of channels.

2.3 Dynamic Channel Assignment

Enormous efforts have been put on the performance analysis of various *dynamic channel assignment (DCA)* algorithms, but most of the results are derived by using simulation or probabilistic modeling. In this thesis, we study the worst-case performance of a well-known DCA strategy, called *greedy channel assignment* in terms of its competitive ratio, for the first time. In this section, we review the experimental work done on DCA.

In contrast to the assumption used in the simple FCA strategy, the traffic distribution is more often non-uniform in the dimensions of time and space. Although the non-uniform allocation technique can be used in FCA to attain high channel efficiency in the presence of spatial variations in traffic, no FCA strategies are able to deal with the temporal traffic variations in the cellular systems.

To overcome this, various DCA schemes have been proposed during the past 20 years. Unlike FCA, there is no definite relationship between channels and cells in DCA. All channels are kept in a central pool and available to every cell. Channels are assigned dynamically to cells when new calls come in and in such a way that co-channel interference is avoided [15, 26, 6, 27]. After a call is completed, its channel is no longer owned by the cell and returned to the central pool.

A channel is eligible for use in any cell as long as the signal interference constraints are satisfied. When a new call comes in a cell, there may be more than one channel available to be assigned to that cell; how to select the assigned channel is the central point of any DCA scheme [6]. The main idea is to evaluate the cost of using each eligible channel, and select the one with the minimum cost. The choice of the cost metric is what differentiates DCA schemes [6].

The cost metrics might be the future blocking probability in the vicinity of the cell, the reuse distance, channel occupancy distribution under current traffic conditions, radio interference measurements of each channel, or the average blocking probability of the system [31]. Based on types of control information used for channel assignment, DCA schemes could be classified either as centralized or distributed schemes.

Centralized DCA Schemes

In centralized DCA schemes, all the call requests are sent to a centralized controller, which also holds the information of the channel usage in each cell. The centralized controller decides which channel from the central pool is assigned to a call for temporary use.

First Available (FA): The simplest of the DCA schemes is the FA strategy. All the channels are indexed. When a cell makes a call request, the first available channel within the reuse distance encountered during a channel search is assigned to the call. The FA strategy needs the minimum system computational time; and, as shown by simulation in [6] for a linear cellular mobile system, it provides an increase of 20 percent in the total handled traffic compared to FCA for low and moderate traffic loads.

Locally Optimized Dynamic Assignment (LODA): In the LODA strategy [35, 34] the selected cost function is a measure on the future blocking probability in the vicinity of the cell in which a call is initiated.

Distributed DCA Schemes

Micro-cellular systems have shown great potential for increasing capacity in high-density communication networks [20, 8]. However, as the cell size becomes smaller, propagation characteristics will be less predictable and network control more complicated than in the present systems. The high cost of control overhead made distributed schemes more attractive for implementation in micro-cellular systems. In distributed DCA schemes, every cell has its own copy of local information and the channel assignment decision is made based on the local information. Two types of local information are used. In cell-based schemes [11, 26], local information about the current channel usage in the cell's vicinity is used. The channel reuse pattern information is updated by exchanging status information between base stations. When traffic is low or moderate, the cell-based scheme is an appealing solution as it provides near-optimum channel allocation. But when traffic is heavy, the expense of excessive exchange of status information between base stations becomes unbearable. The need of inter-cell communication is eliminated in the other distributed solutions which rely on signal strength measurements [30]. A base station uses only local information in its own coverage area. Thus, the system is self-organizing, and channels can be placed or added everywhere, as needed, to increase capacity or improve signal quality in a distributed fashion. At the expense of increased co-channel interference probability with respect to ongoing calls in adjacent cells, these schemes allow fast real-time processing and maximal channel reuse.

Greedy Strategy: The simplest scheme among the interference adaptation DCA schemes is the greedy strategy (also called sequential channel search (SCS) strategy [30]), where all mobile/base station pairs examine all the channels in the same order and the first available channel is chosen. The same strategy for picking the channel to use is also used in [27]. It is expected that this approach will support higher traffic load by packing stations using the same channel at the expense of lower signal quality.

Minimum Signal-to-Noise Interference Ratio (MSIR): In MSIR [30], signal-to-noise information is used to improve system performance. A base station searches for the channel with the minimum interference. Because it tries to assign the channel with least interference to new calls, MSIR has a better signal quality; on the other hand, it is more vulnerable to blocking than SCS. Simulation results have shown that there is a trade-off between the goals of avoiding call blocking and minimizing interference [30].

Dynamic Channel Selection (DCS): Based on the assumption that mobiles are able to measure the amount of interference they experience in each channel, DCS [28] is a fully distributed algorithm for sharing radio resources. Unlike MSIR in which the base stations are responsible for assigning channels, in DCS, each mobile station chooses the base station with the lowest interference probability. A number of parameters are used to calculate the interference probability, such as the received signal power from base stations, the availability of channels, and co-channel interference. In [10], specific models are developed to calculate probabilities of channel availability, desired carrier power, and the CIR for constant traffic load.

Compared to FCA schemes, DCA schemes have more uncertainty involved in channel allocation, which makes it difficult to achieve maximum channel reuse. As a result, DCA schemes do not carry as much traffic at high load. In order to improve the performance of DCA schemes in heavy traffic conditions, channel re-assignment techniques have been suggested [6]. With channel re-assignment, it is possible to switch calls already in process from the channels these calls are using to other channels, with the objective of keeping the distance between cells using the same channel simultaneously to a minimum. Thus, channel reuse is more concentrated, and more traffic can be carried per channel at the expense of more computation efforts put into every allocation decision.

In this thesis, we study the version of DCA given in [27, 30]. We give upper and lower bounds for the competitive (performance) ratio for the online (offline) version of the problem for different values of reuse distance. We also run simulations and study the performance of the algorithm when the number of channels is limited.

2.4 Fixed Preference Channel Assignment

In the *Fixed Preference Channel Assignment (FPA)* strategy [29, 12], each cell is assigned a fixed preference list of all channels. An incoming call in the cell is served by the available channel that comes first in the preference list of the cell. The FPA strategy is a dynamic strategy in the sense that every channel is available for all cells. But because of a cell's preference as to which channels to be used, FPA has less arbitrariness in the channel assignment. What is more, our simulation results in Chapter 5 show that this preference leads to a very strong tendency in a cell using channels from a fixed set of channels. In this sense, FPA is more like FCA.

The principle of FPA is the following. At an initial stage, a good base coloring (using few colors) for the interference graph of the cellular network is found. Suppose this base coloring uses k colors, and suppose there are n channels available to the network, labeled by the numbers from one to n. FPA then uses an (n, k) preference system $R = \{\rho_i | 1 \leq i \leq k\}$.

A *preference list* ρ of order n is a permutation of the numbers $1, 2, \ldots, n$. The rank of a number $j (1 \leq j \leq n)$ in the preference list ρ is defined as $\rho^{-1}(j)$, the position of j in the list ρ. An (n, h) preference system $R = \{\rho_i | 1 \leq i \leq k\}$ is a set of k preference lists of order n.

The preference lists in R are assigned to the cells of the network according to the base coloring of the interference graph. So for each $t \leq k$, the cells that correspond to interference graph vertices of base color t use preference list ρ_t to assign incoming calls. When a cell has a call to serve, the cell will use the first channel in its preference list which is not being used by itself or any of its neighbors. If a channel is released by a terminating call, the other call in the same cell using the last channel on the preference list will be reassigned this channel in order to achieve maximum packing of the channels being used.

The performance of FPA depends largely on the choice of preference system. In [29], Raymond describes a method called Markov allocation, in which the preference lists are chosen heuristically and separately for each case. J.Janssen *et al.* [12] proposed an optimal preference system construction method based on Latin squares.

Even though the above FPA schemes are presented as online algorithms, they allow reassignment of channels of existing calls in order to achieve maximum channel reuse efficiency. In this thesis, we propose an FPA scheme that does not reassign channels, and give a lower bound on its competitive ratio. We also run simulations of the algorithm and study its performance empirically.

Chapter 3

Dynamic Channel Assignment

In this chapter, we study the worst-case performance of the *greedy* strategy described in [28, 30]. Simply put, the greedy strategy uses the minimum numbered channel among those eligible. In this chapter, we study both offline and online versions of the greedy algorithm and give upper and lower bounds on its worst-case performance. Although the greedy strategy is a very simple version of DCA, the results in this chapter also apply to more complicated measurement-based DCA schemes.

3.1 Preliminaries

We define $N_r(v)$ to be all neighbors of v in G^r that have a lower color than that of v. For example, if v is a node with color 3 in G^r, then $N_r(v)$ consists of all neighbors of v in G^r that have color 1 or 2. Given a weighted graph (G, w), for any node v, we define $H_r(v)$ to be $w(v) + \Sigma_{u \in N_r(v)} w(u)$. Given a (partial) assignment of channels to $N_r(v)$, $RC_r(v)$ is defined to be the number of channels assigned to more than one node in $N_r(v)$. Thus $RC_r(v)$ is a measure of the reuse of channels within $N_r(v)$. Finally, we define $\overline{N}_r(v)$ to be all neighbors of v in G^r, and $\overline{H}_r(v)$ to be $w(v) + \Sigma_{u \in \overline{N_r}(v)} w(u)$.

We are now ready to define the greedy algorithm in a more precise manner. Given a weighted hexagon graph (G, w) and a reuse distance r, the algorithm is assumed to have a coloring of G^r using $\chi(G^r)$ colors available. The algorithm proceeds in $\chi(G^r)$ rounds. In round i, each node v with color i and weight $w(v)$ assigns to itself the set of $w(v)$ channels with the lowest numbers that are not used in $N_r(v)$. Notice that all nodes in $N_r(v)$ were assigned channels in previous rounds. This synchronization scheme is given only for ease of analysis: it is not hard to see that the lower bounds obtained for the above algorithm also apply to the greedy DCA algorithms in the literature.

The algorithm described above is an offline algorithm. However, as mentioned in Chapter 2, this can be implemented in an online fashion as follows. At step t, every node v knows its weight $w_t(v)$. In every step, the algorithm proceeds in rounds, and the nodes with color i participate in round i. At every step, a node completely recalculates the channels to be assigned to all its nodes, and the channel assigned to an ongoing call can be changed from time to time. We call this algorithm *G-R*, the greedy algorithm with reassignment. The second version of the greedy algorithm we study is an online algorithm, called *G-NR*, which does not perform reassignments. At step t, each node v knows the number of *new* call arrivals, say $n_t(v)$. The node v then assigns to itself the set of $n_t(v)$ channels with the lowest numbers that are not used at itself or in $\overline{N}_r(v)$. Any synchronization mechanism (including the one described above for the offline algorithm) could be used to prevent conflicts occurring during a single time step.

3.2 Reuse Distance 2

In this section, we study the behavior of the greedy strategy when the reuse distance is 2. We show a tight bound for the performance of the offline version of the greedy algorithm. and a lower bound for the competitive ratio of G-NR. the online greedy algorithm with no reassignments, that matches the upper bound of [5].

3.2.1 Offline Case

For this case, the best known algorithms [22, 25] have performance ratio 4/3. We show that the greedy strategy has performance ratio 5/3. This is a tight bound, as there are weighted hexagon graphs where the greedy algorithm uses 5/3 times the optimal number of channels required.

Let $mc(v)$ be the highest channel used by the vertex v. The following lemma is used frequently in subsequent sections.

Lemma 1 *For the offline greedy algorithm, and for any node v, $mc(v) \leq min\{H_r(v) - RC_r(v), w(v) + max_{u \in N_r(v)} mc(u)\}$.*

Proof: The number of distinct channels used by nodes in $N_r(v)$ is at most $H_r(v) - w(v) - RC_r(v)$. Therefore, v will never use a channel higher than $H_r(v) - RC_r(v)$. Also, if u is a node in $N_r(v)$ that uses the highest channel in v's neighborhood, v will never use more than the next $w(v)$ channels. □

Theorem 1 *For reuse distance 2, the performance ratio of the offline greedy algorithm is $\frac{5}{3}$.*

Proof: Let (G, w) be a weighted hexagon graph, where every node is colored red, blue, or green. We denote $D_2(G, w)$ by D_2. It is easy to see that all red and blue nodes can be assigned channels using the first D_2 channels $C[1, 2, \ldots, D_2]$. In the third round of the algorithm, we assign channels to the green nodes. For any green node v with

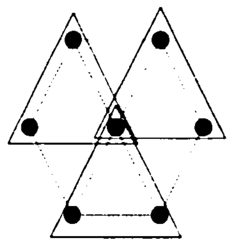

Figure 3.1: $N_2(v) \cup \{v\}$ is covered by 3 cliques.

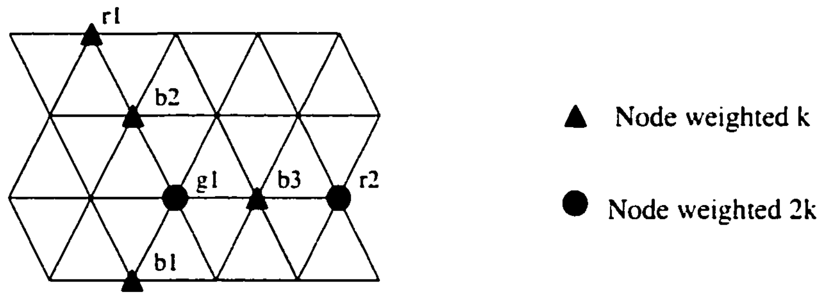

Figure 3.2: An example where the greedy algorithm uses $5D_2(G, w)/3$ channels.

$w(v) \geq \frac{2D_2}{3}$, since $N_2(v)$ and v can be covered by three cliques(as shown in Figure 3.1), $H_2(v) \leq 3D_2 - 2w(v) \leq 3D_2 - \frac{4D_2}{3} = \frac{5D_2}{3}$. Thus, by Lemma 1, $mc(v) \leq H_2(v) \leq \frac{5D_2}{3}$. If instead, $w(v) < \frac{2D_2}{3}$ since we can use the first D_2 colors $C[1, 2, \ldots, D_2]$ to color the red and blue nodes, by Lemma 1, we have $mc(v) \leq w(v) + max_{u \in N_2(v)} mc(u)$ $\leq 2D_2/3 + D_2 = \frac{5D_2}{3}$. Since a green node can never use a channel higher than $5D_2/3$ and D_2 is a lower bound on the number of channels needed, the algorithm has performance ratio at most $5/3$.

We note that $5/3$ is also a lower bound for the performance ratio of the greedy algorithm. For example, in Figure 3.2, the reader can verify that $D_2 = 3k$, and that there is an optimal assignment using $3k$ channels, but the greedy algorithm will use $5k$ channels. Thus, the greedy algorithm uses $5/3$ times the optimal number of channels required. □

Corollary 1 *G-R, the greedy algorithm with reassignments, has competitive ratio* $\frac{5}{3}$ *for reuse distance 2.*

3.2.2 Online Case

For the online case, a trivial upper bound of 3 for the competitive ratio of the greedy algorithm follows from the fact that $\overline{N}_2(v) \cup \{v\}$ can be covered with 3 cliques (see Figure 3.1). We prove a lower bound of 2.5 for this case by constructing a hexagon graph and a sequence of weight vectors.

Theorem 2 *G-NR, the greedy algorithm with no reassignments, has competitive ratio p where $\frac{5}{2} \leq p \leq 3$ for reuse distance 2.*

Proof: We provide a hexagon graph (see Figure 3.3) and a sequence of call arrivals

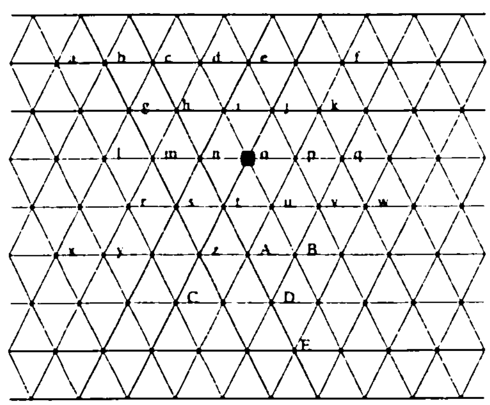

Figure 3.3: Worst case for G-NR for reuse distance 2

and terminations on nodes in the graph so that the greedy algorithm is forced to use 20 channels, while the reader can verify that the optimal offline algorithm needs only 8 channels. Let the pair (α, i) represent:

$$\begin{cases} i \text{ new calls come into node } \alpha & \text{if } i > 0 \\ i \text{ calls finish at node } \alpha & \text{if } i < 0 \end{cases}$$

It is easy to see that the number of calls can easily be multiplied by any $k > 0$ to give arbitrarily large weight vectors. While in principle, the adversary can specify the exact calls that finish, in our example, when reducing the weight, we always reduce it to zero, that is, *all* current calls terminate. Thus there is no need to specify the exact calls that finish in a particular time step. Finally, many elements of the following sequence could be done in parallel; we do not use this optimization for clarity and ease of verification.

The adversary's objective is to make the node o use the channels $C[19, 20]$. This happens only if all the channels $C[1, 18]$ are currently in use in o's neighborhood. Therefore the first goal of the adversary is to make such a situation occur. It is intuitively clear that it is harder to make a neighbor of o use the *high* channels than the *low* channels. Thus we work on the harder task first. We first force the node n to use the channels $C[16, 18]$. To do this, once again, we create a situation where the 6 neighbors of n use all the channels $C[1, 15]$. In particular, the nodes h, o and s will have weight 2 each, and will use the higher channels $C[10, 15]$ and the remaining neighbors of n have weight 3 each, and will use the channels $C[1, 9]$. The value of $D_2[G]$ never exceeds 8.

First we work on the node s. The sequence $(E, 2), (D, 3), (E, -2), (t, 2), (A, 5), (t, -2), (D, -3)$ leads to A using the channels $C[6, 10]$. Next, the sequence $(C, 5), (z, 3), (C, -5), (A, -5)$ leads to z using the channels $C[11, 13]$. Now, the sequence $(x, 3), (y, 3), (x, -3), (\ell, 3), (r, 4), (y, -3), (\ell, -3)$ leads to r using the channels $C[7, 10]$. Increasing the weight of n to 6 leads to n using the channels $C[1, 6]$ since n has no neighbors of positive weight. Now, increasing the weight of s to 2 forces s to use the channels $C[14, 15]$. The sequence $(r, -4), (z, -3)$ ensures that the only nodes with non-zero weight are s and n.

Next we work on the node o. The sequence $(w, 3), (v, 3), (w, -3)$ leads to v using the channels $C[1, 3]$. Next, the sequence $(k, 3), (p, 5), (k, -3), (v, -3), (o, 2), (p, -5)$ leads to the node o using the channels $C[12, 13]$. At this point, o, s, and n are the only nodes with non-zero weight. Next, to make node h use the channels $C[10, 11]$, we employ the sequence $(a, 3), (b, 3), (a, -3), (d, 3), (c, 3), (b, -3), (d, -3), (h, 2)$. Next, the sequence $(n, -6), (c, -3)$ ensures that the only nodes with positive weight are h, o, and s, and they are using the channels $[10, 15]$.

Finally we make the other neighbors of n use the channels $C[1, 9]$. The sequence $(y, 3), (r, 3), (y, -3), (g, 3), (m, 3), (g, -3), (r, -3)$ makes m use the channels $C[7, 9]$. The sequence $(i, 3), (A, 3), (t, 3), (A, -3)$ makes i use the channels $C[1, 3]$ and t use the channels $C[4, 6]$. Then the sequence $(n, 3), (h, -2), (o, -2), (s, -2), (m, -3), (i, -3), (t, -3)$ leads to n using the channels $C[16, 18]$ and being the only node of positive weight in the graph.

Recall that the final goal is to make o use the channels $C[19, 20]$. At this point, we work on the other neighbors of o. First we make the node j use the channels $C[13, 15]$. The sequence $(w, 4), (q, 4), (w, -4), (f, 4), (k, 4), (f, -4), (q, -4)$ leads to k using the channels $C[9, 12]$. Next, $(d, 5), (e, 3), (d, -5), (o, 5), (j, 3), (k, -4), (e, -3), (o, -5)$ achieves the purpose of j using the channels $C[13, 15]$. Now, the sequence $(k, 2), (q, 2), (B, 3), (v, 2), (p, 3), (k, -2), (q, -2), (v, -2)$ leads to p getting the channels $C[7, 9]$. At this point, $(z, 3), (t, 3), (z, -3), (i, 3), (u, 3), (B, -3), (o, 2)$, leads to t getting the channels $C[4, 6]$, i getting the channels $C[1, 3]$, u getting the channels $C[10, 12]$, and finally, o getting the channels $C[19, 20]$ as desired. The reader can verify that the optimal offline algorithm can perform the assignment with 8 channels. □

3.3 Reuse Distance 3

In this section, we study the case when the reuse distance $r = 3$. We show that for the offline case, the greedy strategy has performance ratio at least that of the borrowing strategy given in [7]. We also show that G-NR, the online greedy algorithm with no reassignments, has competitive ratio at least 3.

3.3.1 Offline Case

Feder and Shende have given a borrowing strategy for this case that has performance ratio 7/3 [7]. In particular, for any weighted hexagon graph (G, w), their algorithm uses at most $7D_3(G, w)/3$ channels. We show in this section that there are situations when the greedy algorithm performs worse than this. Also, we show an upper bound on the performance ratio of the greedy algorithm; however, we were unable to prove a tight bound for this case.

For reuse distance 3, the underlying unweighted graph G^3 can be colored with 7 colors. Let G be a weighted hexagon graph and let D_3 denote $D_3(G, w)$. For convenience, we refer to a node with color 1 as a 1-node, a node with color 2 as a 2-node and so on. We show bounds on the largest channel used by nodes of different colors, and use this to show an upper bound on the performance ratio of the greedy algorithm.

Lemma 2 *All the 1-nodes and 2-nodes can be assigned using the first D_3 channel $C[1, 2, \ldots, D_3]$.*

Proof: Straightforward. □

Lemma 3 *For v a 3-node, $mc(v) \leq min\{D_3 + w(v), 3D_3 - 2w(v)\} \leq \frac{5D_3}{3}$.*

Proof: For any 3-node v, the set $\{v\} \cup N_3(v)$ can be covered by three cliques, and also, by Lemma 2, no element of $N_3(v)$ can use a channel higher than D_3. It then follows from an argument similar to the proof for green nodes in Theorem 1 that $mc(v) \leq \min\{D_3 + w(v), 3D_3 - 2w(v)\} \leq \frac{5D_3}{3}$. □

In fact, this is a tight bound, as shown by the example in Figure 3.4. The reader can verify that $D_3 = 3k$, but the greedy algorithm will use $5k$ channels.

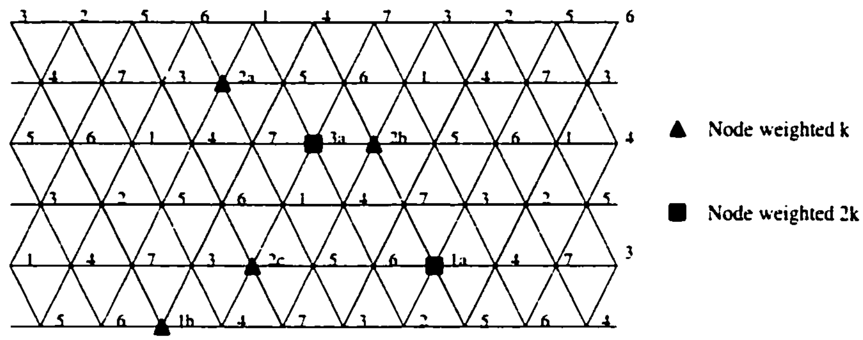

Figure 3.4: An example showing that a 3-node can use $5D_3/3$ channels.

We prove the following general fact that will be useful later.

Fact 1 *Let p be a 3-node in $N_3(v)$ where v is a 4-node, and let $w(p) = D_3 - w(v) - i$. Then $H_3(p) \leq D_3 + 2i$.*

Proof: Since p is a 3-node, $N_3(p)$ can be covered by three cliques, each consisting of a pair of 1- and 2-nodes. Since two of these cliques are also sub-graphs of a clique containing both v and p, their total weights are at most i each (see Figure 3.5 for one such position of p with respect to v, the other positions can be verified by the reader). The third clique in $N_3(p)$ has weight at most $w(v) + i$. Thus, $H_3(p) \leq D_3 + 2i$. $\square$

Lemma 4 *For v a 4-node, $mc(v) \leq \frac{13D_3}{7}$.*

Proof: For any 4-node v, since $\{v\} \cup N_3(v)$ can be covered by three cliques, $H_3(v) \leq 3D_3 - 2w(v)$. So if $w(v) \geq \frac{4D_3}{7}$, by Lemma 1, $mc(v) \leq H_3(v) \leq 3D_3 - 8D_3/7 = \frac{13D_3}{7}$. On the other hand, by Lemma 3, none of the nodes in $N_3(v)$ can use channels higher than $\frac{5D_3}{3}$. Therefore, if $w(v) \leq \frac{4D_3}{21}$, $mc(v) \leq \frac{5D_3}{3} + \frac{4D_3}{21} = \frac{13D_3}{7}$.

It remains to show that $mc(v) \leq \frac{13D_3}{7}$ when $\frac{4D_3}{21} < w(v) < \frac{4D_3}{7}$. Let $w(v) = \frac{4D_3}{21} + k$, where $0 < k < \frac{8D_3}{21}$. Then for any 3-node p in $N_3(v)$, $w(p) \leq \frac{17D_3}{21} - k$. Let $w(p) = \frac{17D_3}{21} - k - i$. We claim that that $mc(p) \leq \frac{5D_3}{3} - k$.

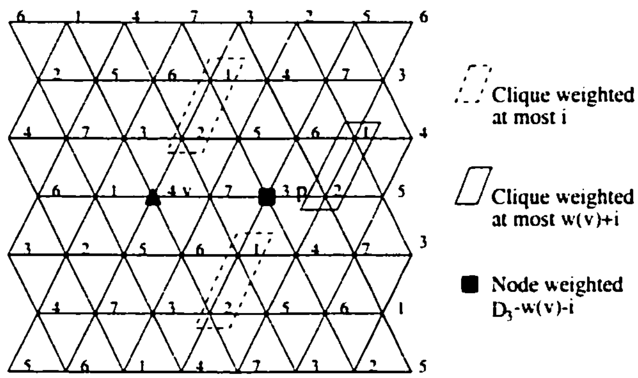

Figure 3.5: Deriving the bound on $H_3(p)$.

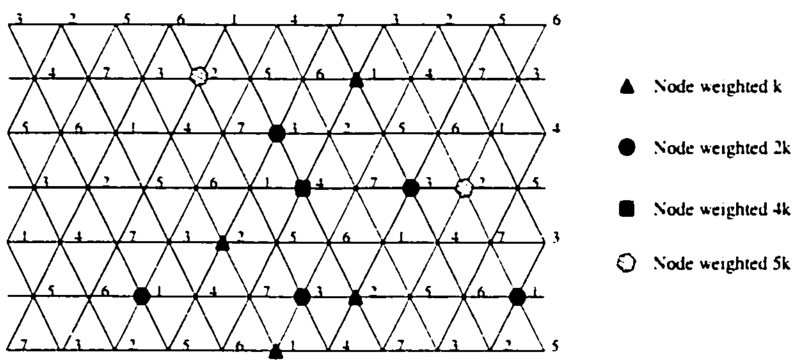

Figure 3.6: An example showing that a 4-node can use $13D_3/7$ channels.

If $i < \frac{D_3}{7}$, by Fact 1, we have $H_3(p) \leq D_3 + 2i \leq 9D_3/7 \leq \frac{5D_3}{3} - k$, and therefore by Lemma 1, $mc(p) \leq \frac{5D_3}{3} - k$. If instead $i \geq \frac{D_3}{7}$, we have $w(p) \leq \frac{2D}{3} - k$. Since none of the 1- and 2-nodes in $N_3(p)$ will use colors higher than D_3, by Lemma 1, $mc(p) \leq D_3 + w(p) \leq \frac{5D_3}{3} - k$.

Since no 3-node in $N_3(v)$ uses a channel higher than $\frac{5D_3}{3} - k$, and any 1- or 2-node in $N_3(v)$ uses channels numbered at most $D_3 < \frac{5D_3}{3} - k$, by Lemma 1, $mc(v) \leq w(v) + \frac{5D_3}{3} - k \leq \frac{13D_3}{7}$. □

The bound in Lemma 4 is a tight bound, as shown by the example in Figure 3.6. The reader can verify that $D_3 = 7k$ but the greedy algorithm uses $13k = 13D_3/7$ channels in this case.

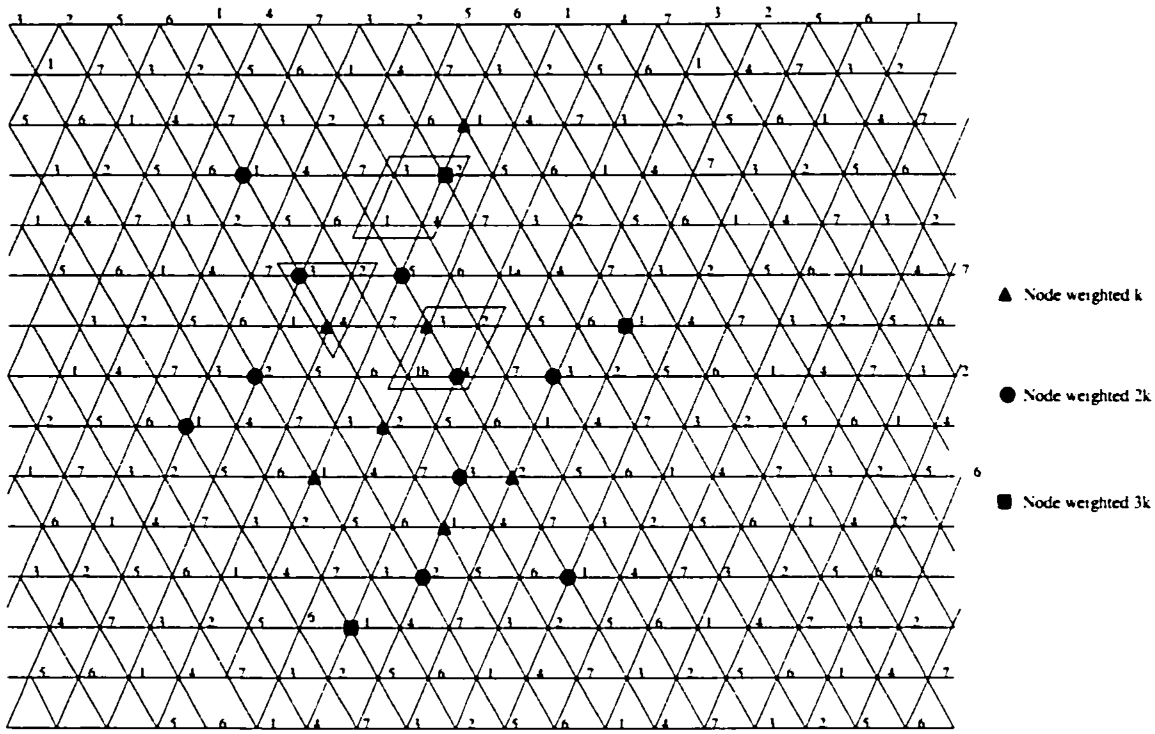

Figure 3.7: An example showing that a 5-node can use $11D_3/5$ channels. Notice that $N_3(v)$ can be covered by 3 cliques and node 1_a. A similar cover using 3 cliques and node 1_b can also be constructed.

Lemma 5 *For v a 5-node, $mc(v) \leq \frac{11D_3}{5}$.*

Proof: For any 5-node v, by Lemmas 2 to 4, since none of its neighbors will use channels higher than $\frac{13D_3}{7}$, if $w(v) \leq \frac{12D_3}{35}$, by Lemma 1, $mc(v) \leq \frac{13D_3}{7} + w(v) = \frac{11D_3}{5}$.

Consider two of the 1-nodes 1_a and 1_b in v's neighborhood (see Figure 3.7). If $w(1_a) \leq w(1_b)$, all of 1_a's channels will also be used by 1_b, and thus $RC_3(v) \geq w(1_a)$. It is easy to verify that $H_3(v) \leq 3D_3 + w(1_a) - 2w(v)$ and therefore by Lemma 1, $mc(v) \leq H_3(v) - RC_3(v) \leq 3D_3 - 2w(v) \leq \frac{11D_3}{5}$ if $w(v) \geq \frac{2D_3}{5}$. Similarly, if $w(1_b) \leq w(1_a)$, we can also show that $mc(v) \leq 3D_3(v) - 2w(v) \leq \frac{11D_3}{5}$.

It remains to show that $mc(v) \leq \frac{11D_3}{5}$ when $\frac{12D_3}{35} < w(v) < \frac{2D_3}{5}$. Let $w(v) = \frac{12D_3}{35} + k$, where $0 < k < \frac{2D_3}{35}$. Then for any 4-node u in v's neighborhood, $w(u) \leq \frac{23D_3}{35} - k$.

Claim 5.1 *$mc(u) \leq \frac{13D_3}{7} - k$, where $0 < k < \frac{2D_3}{35}$.*

Proof: Let $w(u) = \frac{23D_3}{35} - k - i$. Since $N_3(u)$ consists of 3 cliques, and all the nodes in one of them are also neighbors of v, therefore $H_3(u) \le \frac{23D_3}{35} - k - i + 2(\frac{12D_3}{35} + k + i) + i = \frac{47D_3}{35} + k + 2i$. If $w(u) \ge \frac{2D_3}{5}$, we have $k + i < \frac{9D_3}{35}$ and $H_3(u) < \frac{13D_3}{7} - k$, therefore by Lemma 1, $mc(u) \le \frac{13D_3}{7} - k$. If instead $w(u) \le \frac{4D_3}{21} - k$, by Lemmas 1, 2, and 3, $mc(u) \le \frac{5D_3}{3} + w(u) = \frac{13D_3}{7} - k$.

Thus we only need to consider $\frac{4D_3}{21} - k < w(u) < \frac{2D_3}{5}$. Suppose $w(u) = \frac{4D_3}{21} - k + \ell$, where $\ell < \frac{22D_3}{105} + k$. Then for any 3-node p in $N_3(u)$, $w(p) \le \frac{17D_3}{21} + k - \ell$. Let $w(p) = D_3 - w(v) - j = \frac{17D_3}{21} + k - \ell - j$. If $j - k \ge \frac{D_3}{7}$, then $w(p) \le \frac{2D_3}{3} - \ell$ and by Lemma 3, $mc(p) \le \frac{5D_3}{3} - \ell$. It then follows from Lemma 1 that $mc(u) \le \frac{5D_3}{3} - \ell + \frac{4D_3}{21} - k + \ell = \frac{13D_3}{7} - k$ as claimed.

Otherwise if $j - k < \frac{D_3}{7}$, by Fact 1, $H_3(p) \le D_3 + 2j < \frac{9D_3}{7} + 2k$. Thus, by Lemma 1, $mc(p) \le \frac{9D_3}{7} + 2k$. It then follows from Lemma 1 that $mc(u) \le \frac{9D_3}{7} + 2k + \frac{4D_3}{21} - k + \ell = \frac{31D_3}{21} + k + \ell < \frac{59D_3}{35} + 2k < \frac{13D_3}{7} - k$, as required. □

The above claim shows that for any 4-node in v's neighborhood, $mc(u) \le \frac{13D_3}{7} - k$. Indeed, if u is 1, 2, or 3-node in v's neighborhood, $mc(u) \le \frac{5D_3}{3} \le \frac{13D_3}{7} - k$, for all values of k in the range $0 < k < \frac{2D_3}{35}$. It follows as a consequence of Lemma 1 that $mc(v) \le w(v) + \frac{13D_3}{7} - k = \frac{12D_3}{35} + k + \frac{13D_3}{7} - k = \frac{11D_3}{5}$. □

The above bound is tight, as there exists a hexagon graph for which $D_3 = 5k$ but the greedy algorithm uses $11k = 11D_3/5$ channels (see Figure 3.7).

Lemma 6 *For v a 6-node, $mc(v) \le \frac{5D_3}{2}$.*

Proof: Let v be a 6-node, and let u be the node in $N_3(v)$ using the maximum-numbered channel. If u is a 1- or 2-node, then by Lemmas 2 and 1, $mc(v) \le D_3 + w(v) \le 2D_3$. Otherwise, if u is a 3-node, then $w(v) \le D_3 - w(u)$ and by Lemmas 3 and 1, $mc(v) \le D_3 + w(u) + w(v) \le 2D_3$. If u is a 4-node, then $mc(v) \le$

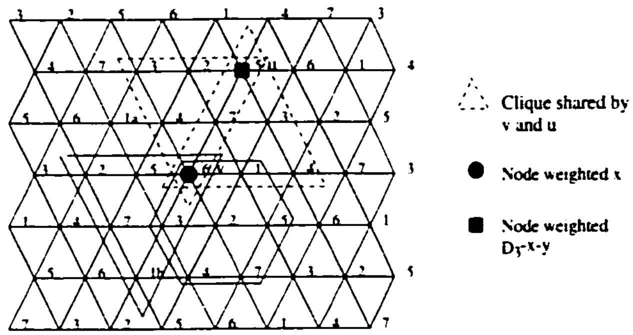

Figure 3.8: Deriving the bound on $H_3(u)$ when $w(1_a) \leq w(1_b)$.

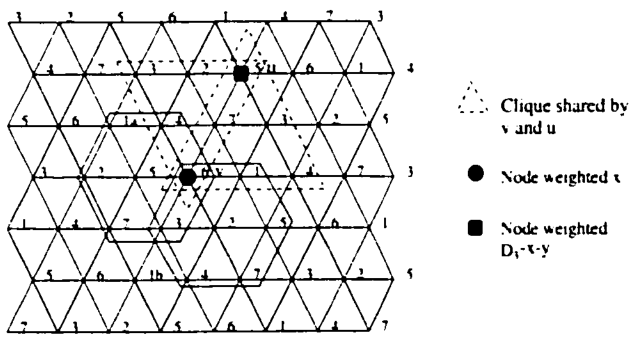

Figure 3.9: Deriving the bound on $H_3(u)$ when $w(1_a) \geq w(1_b)$.

$min\{13D_3/7 + w(v), 3D_3(v) - 2w(v)\} \leq 47D_3/21 \leq 5D_3/2$ as claimed. We consider below the case when u is a 5-node.

Claim 6.1 *Let $w(v) = x$ and $w(u) = D_3 - x - y$ where u is a 5-node.*

1. $mc(v) \leq 3D_3 - 2x + y$.
2. $mc(v) \leq D_3 + 2x + 2y$.
3. $mc(v) \leq (19D_3 + x - 6y)/7$.

Proof: Notice that there are three 5-nodes in $N_3(v)$. In other words, the node u can be in three different positions relative to the node v. One such position is shown in Figures 3.8 and Figure 3.9. The case when $w(1_a) \leq w(1_b)$ is shown in Figure 3.8. It can be seen that $N_3(v) \cup \{v\}$ can be covered by four cliques and the node 1_a. Two of these four cliques also include the node u. Furthermore, since $w(1_a) \leq w(1_b)$, all channels used at 1_a are also reused at node 1_b, thus $RC_3(v) \geq w(1_a)$. Therefore, by Lemma 1, $mc(v) \leq H_3(v) - RC_3(v) \leq 4D_3 + w(1_a) - 3x - (D_3 - x - y) - RC_3(v) \leq 3D_3 - 2x + y$, as claimed. A similar argument applies when $w(1_b) \leq w(1_a)$: see Figure 3.9. Finally, we can show the identical result for the remaining two positions of u. This finishes the proof of (1).

To see (2), observe that $N_3(u)$ consists of 3 cliques of 1-, 2-, 3-, and 4-nodes, in addition to a 1-node, which can always be chosen to be the smaller of nodes 1_a and 1_b in $N_3(u)$ (see Figure 3.7). Further, one of these cliques is such that all nodes in it are neighbors of v, and thus has weight at most y, while the other two cliques have weight at most $x+y$. Thus, using an argument similar to the one in the previous paragraph, we can show that $H_3(u) \leq D_3 + x + 2y$, and thus by Lemma 1, $mc(u) \leq D_3 + x + 2y$. This implies in turn that $mc(v) \leq D_3 + 2x + 2y$.

To show (3), we need to analyze more carefully the maximum channels used by 3 and 4-nodes. In particular, if x_3 is a 3-node with a 4-node neighbor x_4, then $H_3(x_3) \leq 3D_3 - 2w(x_3) - 2w(x_4)$. Therefore, $mc(x_3) \leq 5D_3/3 - 2x_4/3$. We take this into account when analyzing a 4-node p with a 5-node neighbor u. In particular, the maximum channel used by any neighbor of p is at most $5D_3/3 - 2w(p)/3$. Also, $H_3(p) \leq 3D_3 - 2w(p) - w(u)$. It follows from Lemma 1 that $mc(p) \leq 13D_3/7 - w(u)/7$. Recalling that $w(u) = D_3 - x - y$ in this case, we obtain $mc(u) \leq mc(p) + w(u) \leq (19D_3 - 6x - 6y)/7$, and thus, by Lemma 1, $mc(v) \leq (19D_3 + x - 6y)/7$ as claimed. □

The following technical claim establishes a minimum value for the functions obtained above.

Claim 6.2 $min(3D_3 - 2x + y, D_3 + 2x + 2y, (19D_3 + x - 6y)/7) \leq \frac{5D_3}{2}$.

Proof: Assume the claim is not true at $x = a$, $y = b$, that is, all three functions evaluate to greater than $\frac{5D_3}{2}$ for these values. This implies:

$$3D_3 - 2a + b > 5D_3/2 \tag{3.1}$$
$$D_3 + 2a + 2b > 5D_3/2 \tag{3.2}$$
$$(19D_3 + a - 6b)/7 > 5D_3/2 \tag{3.3}$$

Then $(1)+(2) \Rightarrow b > \frac{D_3}{3}$, while $(1)+(3)*14 \Rightarrow b < \frac{7D_3}{22}$, which yields a contradiction. □

Claims 6.1 and 6.2 together establish the lemma. □

Lemma 7 *For v a 7-node, $mc(v) \leq \frac{23D_3}{8}$.*

Proof: For any 7-node v, $N_3(v) \cup \{v\}$ can be covered by four cliques, and so $H_3(v) \leq 4D_3 - 3w(v)$. If $w(v) \geq \frac{3D_3}{8}$, by Lemma 1, $mc(v) \leq H_3(v) \leq 4D_3 - 3w(v) \leq \frac{23D_3}{8}$. On the other hand, by Lemmas 2 to 6, none of the nodes in $N_3(v)$ can use channels higher than $\frac{5D_3}{2}$. So if $w(v) \leq \frac{3D_3}{8}$, by Lemma 1, $mc(v) \leq \frac{5D_3}{2} + w(v) \leq \frac{23D_3}{8}$. □

It is possible to construct a weighted hexagon graph where the greedy algorithm uses $7D_3/3$ channels, while an optimal assignment using D_3 channels exists, as shown in Figure 3.10. In Figure 3.10, the 7-node at the center has to use channel $C[7]$ as channels $C[1, 2, ..., 6]$ are being used by its neighbors. The optimal assignment, however, needs only 3 channels. Thus, the performance ratio of the greedy algorithm is at least 7/3. However, there is a weighted hexagon graph (G, w) where $D_3(G, w) = 5k$ and the greedy algorithm uses $12k = 12D_3/5$ channels, as shown in Figure 3.11[1]. The reader can verify the assignment given by the greedy algorithm.

[1]However, this graph does not appear to have an assignment with fewer than $6k$ channels.

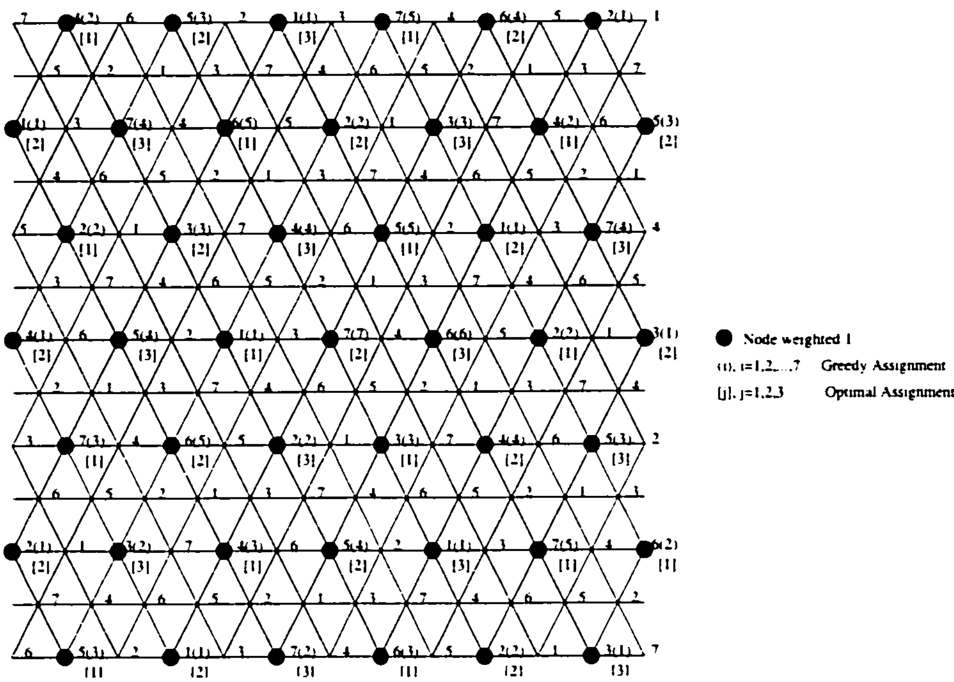

Figure 3.10: An example showing that greedy algorithm uses 7 channels while optimal assignment needs only 3 channels.

The following theorem is a consequence of Lemmas 2 to 7 and the discussion in the paragraph above:

Theorem 3 *For reuse distance 3:*

1. *The offline greedy algorithm has performance ratio p where* $\frac{7}{3} \leq p \leq \frac{23}{8}$.
2. *There is a weighted hexagon graph* (G, w) *such that the offline greedy algorithm uses* $12D_3(G, w)/5$ *channels.*

Corollary 2 *G-R, the online greedy algorithm with reassignments, has competitive ratio p where* $\frac{7}{3} \leq p \leq \frac{23}{8}$ *for reuse distance 3.*

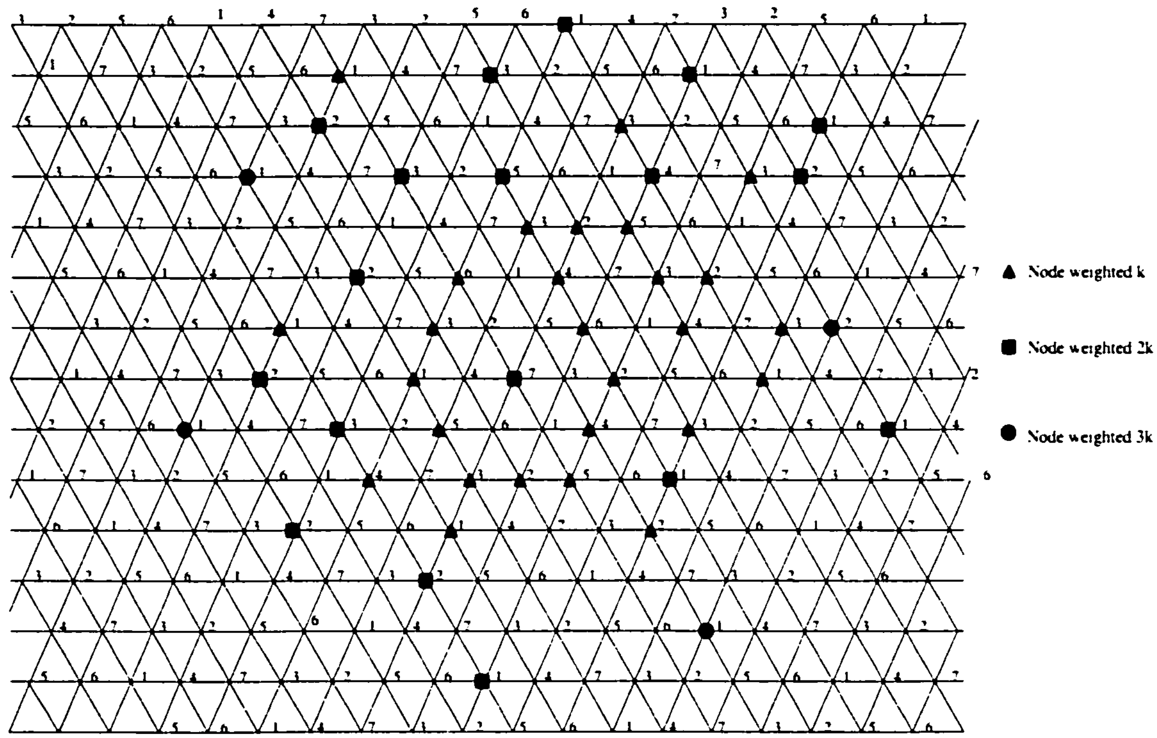

Figure 3.11: An example showing that a 7-node can use $12D_3/5$ channels.

3.3.2 Online Case

For this case, a simple modification of the cluster partitioning strategy given in [16] has competitive ratio 4. We show here that the greedy algorithm has competitive ratio between 3 and 4.

Theorem 4 *G-NR, the greedy algorithm with no reassignments, has competitive ratio p where $3 \leq p \leq 4$ for reuse distance 3.*

Proof: The upper bound on p follows simply from the fact that for any node v, $\overline{N}_3(v)$ is covered by 4 cliques, each with weight at most D, which implies that $mc(v) \leq \overline{H}_3(v) \leq 4D$. To show the lower bound, we construct a hexagon graph and a series of call arrivals and departures so that the algorithm is forced to use 9 channels while the optimal offline algorithm needs only 3 channels. The proof is similar to that of Theorem 2, but the sequence of call arrivals and terminations as well as the graph

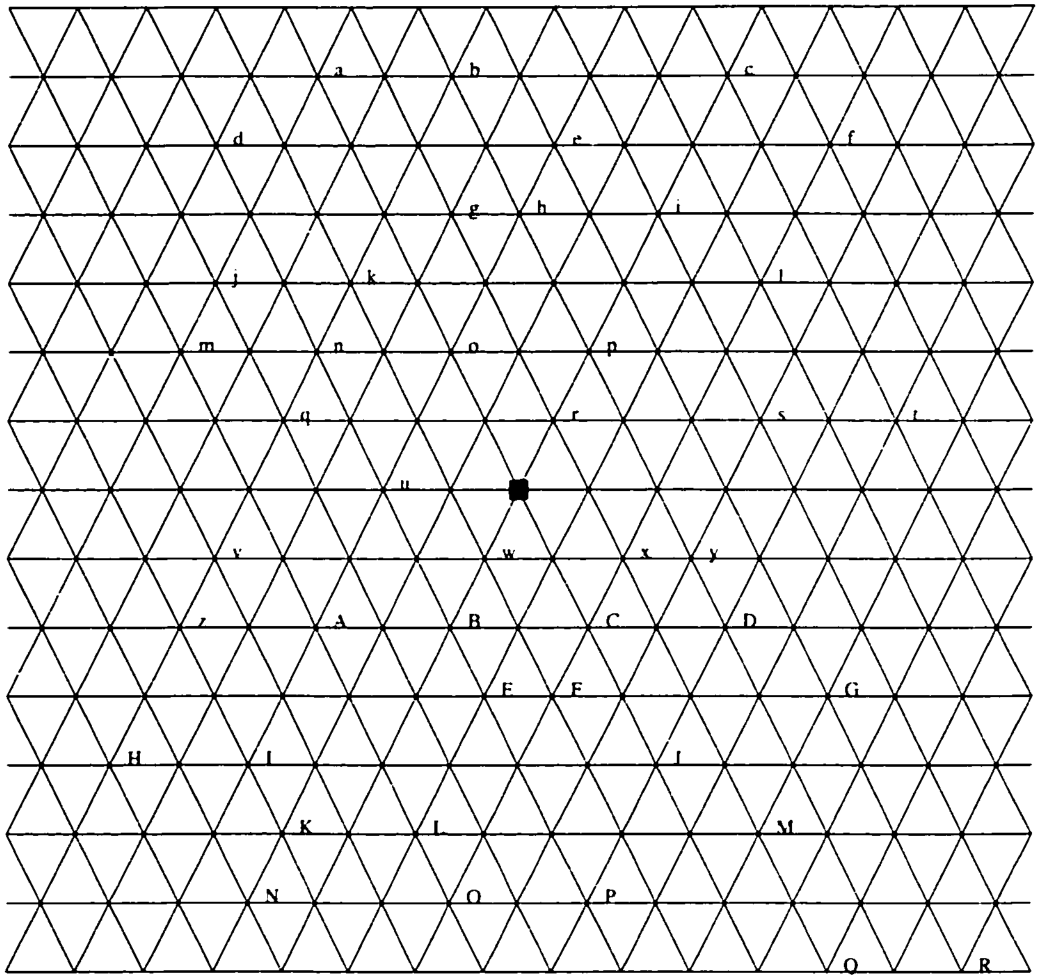

Figure 3.12: Graph in which nodes C and o are forced to use channels 7 and 8 respectively. See Tables 3.1 and 3.2.

used are much larger. In what follows, we use $(x, w, [c_1, c_2])$ to denote the arrival of w new calls at node x, which increases the weight of node x to w from 0. This in turn forces the greedy algorithm to use the contiguous channels c_1 to c_2 at node x. The notation $\not{x}$ denotes the removal of all calls at node x, thus reducing the weight to 0, and causing the release of all channels currently being used at node x. We note that many of the steps in this sequence can be performed in parallel, thus shortening the sequence, as well as the number of steps required. We give this longer sequence here for clarity and ease of verification.

Since the number of nodes involved is quite large, we break up the sequence in three sequences, given in Tables 3.1, 3.2 and 3.3. The first sequence, given in Table 3.1, is intended to make the node o use the channel 8. The corresponding graph is given in Figure 3.12. In order to make this happen, we have to first make the neighbors of o use all channels $C[1,7]$. These goals are listed in the first column on Table 3.1. These in turn lead to other sub-goals listed in the second column. Similarly, Table 3.2 in conjunction with Figure 3.12 describes the sequence to force node C to use channel 7 given that node o is already using channel 8. Finally, Table 3.3 in conjunction with Figure 3.13 gives the sequence to force node p to use channel 9 given that nodes o and C are using channel $C[8]$ and $C[7]$. The reader can also verify that the optimal offline algorithm can perform the assignment with 3 channels. □

3.4 Arbitrary Reuse Distance

In this section, we consider the problem of channel assignment with respect to an arbitrary reuse distance r. We show a lower bound on the performance of any algorithm, and analyze the worst-case performance of FCA, the greedy strategy as well as a straightforward borrowing strategy.

3.4.1 Lower Bounds

Lemma 8 *For any reuse distance $r \geq 3$, there are hexagon graphs G such that G^r contains a 5-cycle as an induced subgraph.*

Proof: If r is odd, we choose $v_1, v_2, \ldots, v_5$, at coordinates $(0,0)$, $(r-1,0)$, $(r-1,r-1)$, $(0, \frac{3r-3}{2})$, and $(-\frac{r-1}{2}, r-1)$ respectively. Otherwise, we choose them at coordinates $(0,0), (r-1,0), (r-1,r-1), (0, \frac{3r}{2}-2)$, and $(-\frac{r}{2}, r-1)$ respectively. It can be easily

Goal	Sub-goal	Sequence
$(h, 1, [7])$		generate node h's neighborhood
	$(o, 2, [1, 2])$	$(e, 1, [1]), (f, 1, [1]), (c, 1, [2]), (o, 2, [1, 2])$
	$(i, 2, [5, 6])$	$(p, 1, [3]), (t, 1, [1]), (s, 2, [2, 3]), (l, 1, [4]), (i, 2, [5, 6])$
	$(b, 2, [3, 4])$	$(d, 1, [1]), (a, 1, [2]), (b, 2, [3, 4])$
		clean up neighborhoods of i, b, and o
		$\cancel{d}, \cancel{a}, \cancel{e}, \cancel{c}, \cancel{p}, \cancel{f}, \cancel{l}, \cancel{s}, \cancel{t}$
		raise h's weight
		$(h, 1, [7])$
		clean up h's neighborhood
		$\cancel{i}, \cancel{b}, \cancel{o}$
$(u, 1, [6])$		generate node u's neighborhood
	$(A, 1, [5])$	$(z, 2, [1, 2]), (H, 1, [3]), \cancel{z}, (v, 1, [1])$
		$(N, 2, [1, 2]), (I, 1, [4]), (F, 1, [1]), (B, 2, [2, 3]), (A, 1, [5])$
	$(q, 2, [3, 4])$	$(d, 1, [1]), (j, 1, [2]), (q, 2, [3, 4])$
	$(o, 2, [1, 2])$	$(o, 2, [1, 2)$
		clean up neighborhoods of A, q, and o
		$\cancel{d}, \cancel{j}, \cancel{o}, \cancel{H}, \cancel{I}, \cancel{N}, \cancel{B}, \cancel{F}$
		raise u's weight
		$(u, 1, [6])$
		clean up u's neighborhood
		$\cancel{A}, \cancel{q}, \cancel{o}$
$(k, 1, [5])$		generate k's neighborhood
	$(q, 2, [3, 4)$	$(v, 1, [1]), (d, 1, [1]), (j, 1, [2]), (q, 2, [3, 4])$
	$(o, 2, [1, 2])$	$(o, 2, [1, 2])$
		clean up neighborhoods of q and o
		$\cancel{d}, \cancel{v}, \cancel{j}$
		raise k's weight
		$(k, 1, [5])$
		clean up k's neighborhood
		$\cancel{q}, \cancel{o}$
$(r, 1, [4])$		$(D, 2, [1, 2]), (x, 1, [3]), (p, 2, [1, 2]), (r, 1, [4])$
		$\cancel{D}, \cancel{x}, \cancel{p}$
$(n, 1, [3])$		$(m, 2, [1, 2]), (n, 1, [3]), \cancel{m}$
$(p, 1, [2])$		$(o, 1, [1]), (p, 1, [2]), \cancel{o}$
$(g, 1, [1])$		$(g, 1, [1])$
$(o, 1, [8])$		$(o, 1, [8])$
		$\cancel{h}, \cancel{u}, \cancel{k}, \cancel{r}, \cancel{n}, \cancel{p}, \cancel{g}$

Table 3.1: Sequence of calls as a result of which node o uses channel 8. See Figure 3.12.

Goal	Sub-goal	Sequence
$(J, 2, [5, 6])$	$(M, 1, [4])$	$(R, 2, [1, 2]), (Q, 1, [3]), (G, 2, [1, 2]), (M, 1, [4])$
	$(D, 1, [3])$	$(D, 1, [3])$
	$(P, 1, [2])$	$(O, 1, [1]), (P, 1, [2])$
	$(F, 1, [1])$	$(F, 1, [1])$
		$\cancel{R}, \cancel{O}, \cancel{Q}, \cancel{G}$
		$(J, 2, [5, 6])$
		$\cancel{M}, \cancel{D}, \cancel{P}, \cancel{F}$
$(E, 2, [3, 4])$	$(L, 1, [2])$	$(O, 1, [1]), (L, 1, [2])$
	$(w, 1, [1])$	$(w, 1, [1])$
		$(E, 2, [3, 4])$
		$\cancel{O}, \cancel{w}, \cancel{L}$
$(y, 2, [1, 2])$		$(y, 2, [1, 2])$
$(C, 1, [7])$		$(C, 1, [7])$
		$\cancel{J}, \cancel{E}, \cancel{y}$

Table 3.2: Sequence of calls as a result of which node C uses channel 7, given that node o is already using channel 8. See Figure 3.12.

Goal	Sub-goal	Sequence
$(i,1,[6])$	$(j,1,[4])$	$(e,2,[1,2]),(d,1,[3]),(s,2,[1,2]),(j,1,[4])$
	$(b,1,[2])$	$(a,1,[1]),(b,1,[2])$
	$(c,1,[5])$	$(c,1,[5])$
	$(u,1,[3])$	$(y,2,[1,2]),(u,1,[3])$
	$(p,1,[1])$	$(p,1,[1])$
		$(i,1,[6])$
		$\cancel{e}, \cancel{y}, \cancel{j}, \cancel{b}, \cancel{c}, \cancel{u}, \cancel{p}, \cancel{a}, \cancel{d}, \cancel{s}$
$(t,1,[5])$	$(m,1,[2])$	$(h,1,[1]),(m,1,[2]), \cancel{h}$
	$(v,1,[3])$	$(A,1,[1]),(v,1,[3])$
	$(x,1,[4])$	$(D,1,[2]),(x,1,[4]), \cancel{A}, \cancel{D}$
	$(p,1,[1])$	$(p,1,[1])$
		$(t,1,[5])$
		$\cancel{m}, \cancel{v}, \cancel{x}, \cancel{p}$
$(n,1,[4])$	$(f,1,[1])$	$(f,1,[1])$
	$(l,1,[2])$	$(l,1,[2])$
	$(v,1,[3])$	$(z,1,[1]),(v,1,[3]),\cancel{z}$
		$(n,1,[4])$
		$\cancel{v}, \cancel{l}, \cancel{f}$
$(q,1,[3])$		$(r,2,[1,2]),(q,1,[3]),\cancel{r}$
$(w,1,[2])$		$(B,1,[1]),(w,1,[2]), \cancel{B}$
$(g,1,[1])$		$(g,1,[1])$
$(p,1,[9])$		$(p,1,[9])$

Table 3.3: Sequence of calls as a result of which node p uses channel 9, given that nodes o and C are already using channels 8 and 7 respectively. See Figure 3.13.

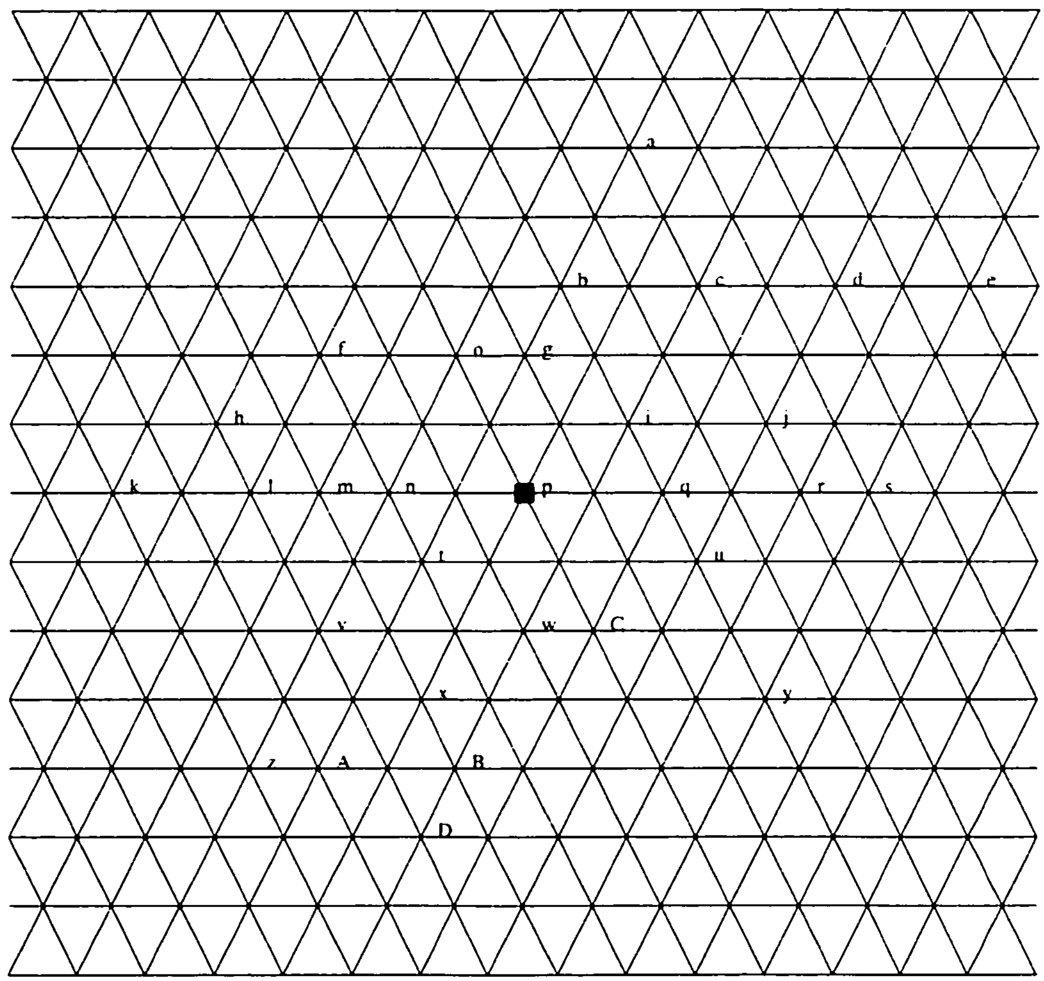

Figure 3.13: Graph in which node p is forced to use channel 9, assuming that nodes o and C are using channels 8 and 7 respectively. See Table 3.3.

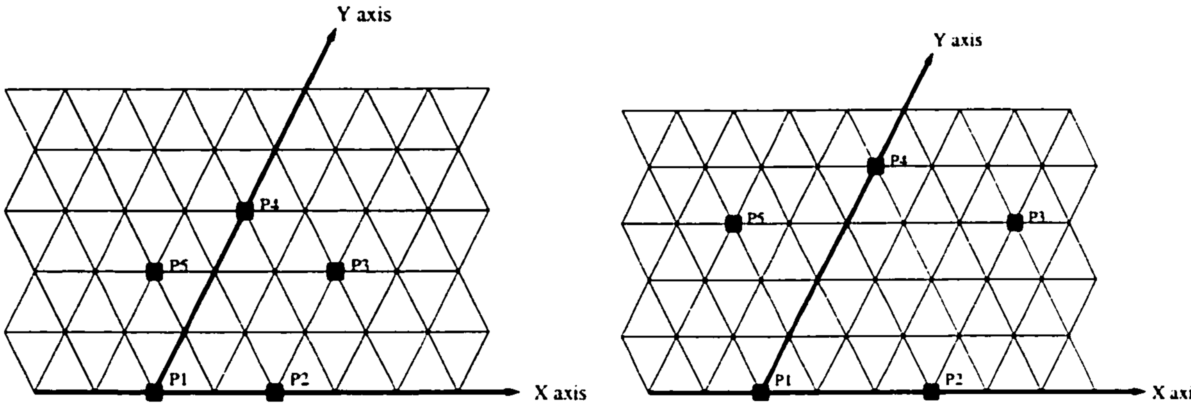

Figure 3.14: (a) 5-cycle for reuse distance 3, and (b) 5-cycle for reuse distance 4.

verified that for any two nodes (v_i, v_j), $d(v_i, v_j) \leq r$ if and only if $i = j \bmod 5 + 1$. For example, in Figure 3.14(a), given reuse distance 3, the 5-cycle consists of the five nodes at $(0,0), (2,0), (2,2), (0,3), (-1,2)$, and in Figure 3.14(b), given reuse distance 4, the 5-cycle consists of the five nodes at $(0,0), (3,0), (3,3), (0,4), (-2,3)$. □

Theorem 5 *For any reuse distance $r \geq 3$, there exists a weighted hexagon graph (G, w) such that any algorithm for channel assignment must use at least $5D_r(G, w)/4$ channels.*

Proof: Let G be a hexagon graph that contains a 5-cycle as an induced subgraph (its existence is confirmed by Lemma 8). We assign every node in the 5-cycle with weight k. Then $D_r(G, w) = 2k$. Further, since $5k$ calls need to be served, and each channel can be used by at most 2 nodes, the optimal number of channels required is $\frac{5k}{2} = 5D_r(G, w)/4$. □

3.4.2 Offline Case

In this section, we analyze the performance of a simple borrowing strategy, along the lines of the algorithm for reuse distance 3 in [7]. The following lemma was derived independently in [23] and [32]. We give a simple construction below for convenience.

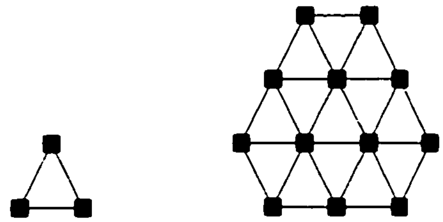

Figure 3.15: Maximum-sized cliques of hexagon graph with reuse distance (a) 2 and (b) 4.

Lemma 9 ([23, 32]) *Given G a hexagon graph, and reuse distance r,*

$$\omega(G^r) = \chi(G^r) = \begin{cases} \frac{3r^2}{4} & \textit{if } r \textit{ is even} \\ \frac{3r^2+1}{4} & \textit{otherwise} \end{cases}$$

Let G be the infinite triangular lattice. We construct the maximum-sized clique in G^r by finding nodes in G that are at distance at most $r-1$ from each other. If r is even, we build *layers* moving outwards from a triangle. The first layer consists of 3 nodes, the second layer consists of 9 nodes, and the $\frac{r}{2}^{th}$ layer, which is the outermost layer, consists of $3 + 6(\frac{r}{2} - 1) = 3r - 3$ nodes. It is easy to verify that this set of nodes comprises a clique, and in fact, a maximum-sized clique. The total number of nodes in the clique is $3 + 9 + ... + (3 + 6(\frac{r}{2} - 1)) = \frac{3r^2}{4}$. For example, Figure 3.15(a) shows that $\omega(G^2) = 3$, and Figure 3.15(b) shows that $\omega(G^4) = 12$. It is also easy to construct a tiling of the infinite triangular lattice using these cliques, thereby showing that the chromatic number equals the clique number.

If instead r is odd, in building the maximum-sized clique we move outwards from a single node. The first layer consists of 1 node, the second layer consists of 6 nodes, the third layer consists of 12 nodes, and the $\frac{r+1}{2}^{th}$ layer consists of $3r-3$ nodes. Therefore the total number of nodes in the clique is $\frac{3r^2+1}{4}$. For example, Figure 3.16(a) shows that $\omega(G^3) = 7$, and Figure 3.16(b) shows that $\omega(G^5) = 19$. Once again, these cliques can be used to tile the triangular lattice, showing that the chromatic number equals the clique number.

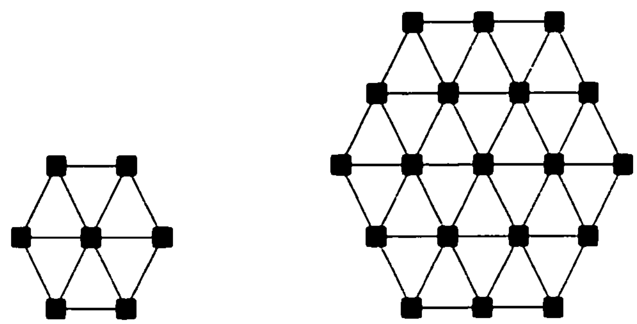

Figure 3.16: Maximum cliques of hexagon graph with reuse distance (a) 3 and (b)5.

Theorem 6 *For any weighted hexagon graph (G, w) and any reuse distance $r > 1$, there is an algorithm that has performance ratio $\frac{18r^2}{3r^2+20}$ if r is even, and $\frac{18r^2+6}{3r^2+21}$ when r is odd.*

Proof: The algorithm uses $\ell\chi(G^r)$ channels (where ℓ will be specified later), and partitions them into $\chi(G^r)$ sets of ℓ channels each. A node v is called *heavy* if $w(v) > \ell$ and *light* otherwise. Each node v of color i in G^r assigns the smallest $min\{w(v), \ell\}$ channels from channel set i. At this point, all light nodes have received enough channels and drop out. Any remaining heavy node now borrows any channel that is unused among its neighbors in G^r.

For any heavy node v, it is easy to see that $H_r(v) \leq w(v) + 6(D_r(G, w) - w(v)) = 6D_r(G, w) - 5w(v)$. For $\ell = \frac{6D_r(G,w)}{\chi(G^r)+5}$, since $w(v) > \ell$, $H_r(v) \leq 6D_r(G, w) - 5\ell = \chi(G^r)\ell$. This means that v has sufficient channels to borrow and to complete its assignment.

From Lemma 9, we know the $\chi(G^r)$ is $\frac{3r^2}{4}$ for even r and $\frac{3r^2+1}{4}$ for odd r, so $H_r(v) \leq \frac{18r^2}{3r^2+20}D_r(G, w)$ for even r and $\frac{18r^2+6}{3r^2+21}D_r(G, w)$ for odd r. It follows from Lemma 1 that $mc(v) \leq H_r(v)$ and thus the performance ratio is at most $\frac{18r^2}{3r^2+20}$ for even r and $\frac{18r^2+6}{3r^2+21}$ for odd r. □

We note that this ratio is always lower than 6 for any bounded value of r. For the particular reuse distances of 2, 3, and 4, this algorithm gives us performance ratios of

2.25, 3.5, and 4.24 respectively. However, the cluster partitioning algorithm described in [16] has a better performance ratio for all values of reuse distance greater than 3, in addition to having the advantage of not requiring reassignment of calls in the online case.

3.4.3 Online Case

The only previously known online algorithm for arbitrary reuse distance is the cluster partitioning [16], which has performance ratio 3 for reuse distance 2 and performance ratio 4 for reuse distance ≥ 3. In this section, we give straightforward bounds on the competitive ratio of FCA and the greedy algorithm, for arbitrary values of reuse distance r.

Theorem 7 *For arbitrary reuse distance $r \geq 3$, the fixed assignment algorithm has performance ratio p where:*

$$p = \chi(G^r) = \begin{cases} \frac{3r^2}{4} & \text{if } r \text{ is even} \\ \frac{3r^2+1}{4} & \text{otherwise} \end{cases}$$

Proof: Follows from Lemma 9.

□

Theorem 8 *For arbitrary reuse distance $r \geq 3$, the online greedy algorithm has performance ratio p where:*

$$\frac{5}{4} \leq p \leq 6$$

Proof: The online greedy algorithm needs no more than $6D_r$ channels for its use. For any node v, its neighborhood $N_r(v)$ can be covered with at most 6 cliques, therefore, the total number of calls in the v's neighborhood is at most $6D_r$. When node v has a new call come in, there are at most $6D_r - 1$ ongoing calls in v's neighborhood. The worst case happens when all the ongoing calls are using different channels, this is, at most $6D_r - 1$ channels are being used. However, v can still find an unused channel. The lower bound follows from Theorem 5. $\square$

Chapter 4

Fixed Preference Assignment

In this chapter, we study the worst-case performance of the FPA strategy described in Section 2.4. In this strategy, the nodes are partitioned into different sets based on their color in G_r, and different nodes of different colors assign channels in order from different preference lists. We briefly describe the previous results on the offline version. We describe an online version and give a lower bound on its performance for reuse distance 2. Straightforward bounds are also given for reuse distance 3.

4.1 Reuse Distance 2

In this section, we describe bounds on the performance of the offline and online versions of FPA.

4.1.1 Offline Case

Theorem 9 *[12] For reuse distance 2. the performance ratio of the offline FPA algorithm is $\frac{3}{2}$.*

We give a proof here for completeness. Let the clique bound be D_2. and the number of channels available be $3D_2/2$. We argue that a valid assignment will be made by FPA. Divide the available $3D_2/2$ channels $C[1.2.\ldots.3D_2/3]$ into 3 equal sized sets: *red*:$C[1.2.\ldots.D_2/2]$. *blue*:$C[D_2/2+1.D_2/2+2.\ldots.D_2]$. and *green*:$C[D_2+1.D_2+2.\ldots.3D_2/2]$. A red node takes as many red channels as it needs. starting from the first. and if it still needs more channels. takes blue channels starting from the end. Similarly, blue nodes borrow from red if necessary. and green nodes borrow from blue. Suppose this assignment of channels causes a conflict between a green node and a red node that are neighbors. Then their combined weight must be greater than D_2, a contradiction. Thus. the assignment is conflict-free. and the algorithm has performance ratio 3/2.

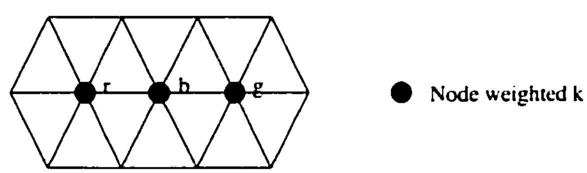

Figure 4.1: Worst case for offline FPA for reuse distance 2

It can be shown that 3/2 is also the lower bound for the performance ratio of the FPA algorithm. In Figure 4.1. FPA will use $3k$ channels while the optimal algorithm uses only $2k$ channels.

4.1.2 Online Case

First, we describe an online version of FPA. A 3-cell reuse pattern is used (see Figure 2.1). Based on their base colors, all the cells are grouped into three sets: *red*, *blue*, and *green*. Also, all the available channels are numbered and equally divided into sets, namely *red*, *blue*, and *green*, each assigned as the primary channels to the cells with the same base color. When a red cell has a call come in, it will try to use the first available channel from its primary channel set. If all its primary channels are being used(either by itself or one of its six neighbors), it will borrow an idle channel from blue neighbors, searching from the first blue channel to the last one. If the searching in the blue set fails again, the red cell will go for the first available green channel. If all the channels are being used, the call is blocked. Similarly, the blue cells and green cells will search for available channels in all three channel sets, but following different orders. Blue cells will follow the order of blue-green-red while green cells' order is green-red-blue.

For the online case, a upper bound of 3 for the performance ratio of the FPA follows from the fact that the maximum weight of a node is D_2, and with D_2 channels in every primary channel set, every node's demand will be satisfied with its primary channels. In fact, the algorithm reduces to FCA when there are 3D channels available.

We prove a lower bound of 2.25 for this case by constructing a hexagon graph and a sequence of weight vectors.

Theorem 10 *The online FPA algorithm with no reassignments, has competitive ratio* p *where* $p \geq 2.25$ *for reuse distance* 2.

Proof: Assume there are 18 channels available, which are divided into three sets: *red*: $C[1, 2, \ldots, 6]$, *blue*: $C[7, 8, \ldots, 12]$, and *green*: $C[13, 14, \ldots, 18]$. The adversary's objective is to make the node r_7 use the channels $C[16, 18]$. This happens only if all the channels $C[1, 15]$ are currently in use in r_7's neighborhood.

To achieve that, the adversary's strategy is to assign the higher numbered channels first and then the lower numbered channels to r_7's neighbors. Therefore, the adversary

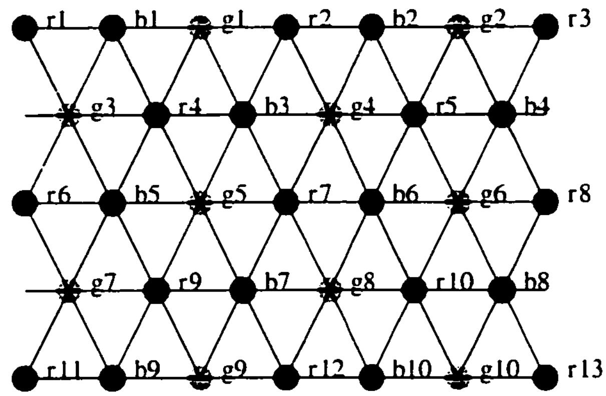

Figure 4.2: Worst case for online FPA for reuse distance 2

first makes node b_6 use the channels $[4,6]$, b_7 use the channels $[7,9]$. This is achieved by the sequence $(r_9,7),(r_{12},8),(r_9,-6),(r_{12},-7),(b_7,7),(b_7,-4),(b_6,6),(b_6,-3)$.

Next, the adversary makes g_5 use the channels $[14,15]$, g_8 use the channels $[12,13]$ and g_4 use channels $[10,12]$ by the sequence $(r_4,1),(g_5,5),(g_5,-3),(r_4,-1),(g_8,4),(g_8,-2),(g_4,5),(g_4,-3)$.

Now, raise the weight of b_3 to 3 and it will use the channels $[1,3]$.

Finally, raise the weight of r_7 to 3 and it has to use the channels $[16,18]$ while the optimal offline algorithm will never need more than 8 channels during the whole process. □

4.2 Reuse Distance 3

J.Janssen *et al.* [12] proposed an optimal offline FPA algorithm in which preference lists are constructed using Latin squares. They prove the following:

Theorem 11 *[12] For reuse distance 3, the offline FPA algorithm has performance* *ratio p:*

$$3.5 \leq p \leq 7$$

As reassignments are used in the offline FPA algorithm to allow maximum channel reuse. its performance ratio serves as the lower bound of the competitive ratio of the online FPA algorithm proposed in this thesis. which does not perform reassignment. On the other hand. all FPA algorithms share the upper bound given by Lemma 9 in Chapter 3. which is 7 for reuse distance 3.

Theorem 12 *For reuse distance 3. the online FPA algorithm has competitive ratio p where:*

$$3.5 \leq p \leq 7$$

Chapter 5

Simulation Experiments and Results

Besides the worst case analysis of the channel assignment algorithms, the average case performance of the algorithms are also of interest. In this chapter, we study the performance of the channel assignment algorithms under different traffic models.

5.1 Techniques of Performance Evaluation

In this section, we give a brief introduction to the commonly used techniques in performance evaluation.

Measurement, analytical modeling, and simulation are three most commonly used techniques for performance evaluation. The development stage of the system, cost, accuracy and flexibility are among the most important factors to be considered when choosing which evaluation technique to be used.

Measurement

Measurements are possible if something similar to the proposed system is available, for example, when designing an improved version of a product. Measurements are best done in hardware although sometimes they could also be done in software or in a hybrid manner. It is the most costly technique as it requires real equipment, instruments, and time. Unlike analytical modeling and simulation, which usually involves many assumptions about the system being studied, measurements can be used to evaluate the system performance in the real world, minimizing the risk of oversimplification. Measurement's importance also lies in the fact that it provides the basis for the many assumptions used in analytical modeling and simulation. To minimize the effects of uncontrolled/uncontrollable factors in a measurement, the data collected from measurements must be analyzed using statistical techniques before we can draw any meaningful conclusions. Careful design of a measurement is very important because of several reasons. Firstly, measurement may not give accurate results due to the uniqueness of environmental parameters such as system configuration, characteristic of workload, and time of measurement, in every measurement experiment. Secondly, the parameters used in the measurements may not represent the full range of the variables. In order to capture the major effects at a lower cost, one must carefully select which parameters are to be measured, how they should be measured, and what input value combinations are used for the parameters.

Analytical Modeling

Analytical modeling can be used at any stage of the system development and usually it has the lowest cost in terms of equipment and manpower requirements. There are situations where the system is not available for measurement. For example, for a proposed new concept, analytical modeling and simulation are the only choices for performance evaluation. Analytical modeling can be used to study the system behavior at any desired level of detail by constructing a mathematical model and solving it. Comparing different alternatives and finding the optimal parameter value are the most important goals of every performance study. Although some simplifications about the system are used in the model in order to make mathematical analysis possible, analytical models can still provide us good insight into the effects on the system performance of various parameters and their interaction. Although a more complex model as a result of fewer assumptions may provide more detailed information about the system performance, a simple model is more desirable because it can be solved easily and sometimes still provides accurate results. Also, simpler models are more robust to changes in parameter values, easier to understand and modify. Unlike measurement or simulation, in which the results of each experiment are different due to the variance of parameter values and only characterize the system behavior for the range of input parameters covered, analytical modeling provides more predictive results than those obtained from measurement or simulation. The main difficulty of the analytical modeling is that the domain of tractable models is rather limited. When the objective is to study the system behavior in great detail, the analytical model will become too complex to be manageable.

Simulation Modeling

There is growing interest in using computer simulation to explore issues in science and engineering. In addition to its use in verifying the correctness of system design, simulation is also widely used as a tool of performance prediction and performance optimization. Simulation is the mimicking of the operation of a real system, it involves designing a model for the behavior of the system, represented by a computer program, generating an appropriate abstraction of the workload driving it, as well as gathering and analyzing the execution output.

The major advantage of simulation over the other two techniques is its flexibility, almost any behavior of the system can be simulated. Simulation can be used to study the system performance at any stage of the development. During the early design stage, if the system to be investigated is not available, a simulation model provides an easy way to predict the performance or compare several design alternatives. Further, even if a system is available for measurement, a simulation model may still be a preferred approach over measurement because it allows the alternatives to be compared under a wider variety of environment parameter values. Unlike analytical models, which often require many assumptions and are too restrictive for most real-world systems, simulation modeling places few restrictions on the systems under study. However, there are some important issues that must be considered in simulation:

• It must be decided at what level of detail the simulation is to be done. In general, the simulation can be done at several levels of detail, depending on the objectives of the study. A simulation with too little detail is easier to implement and verify, but it can provide misleading or incorrect results. A more detailed model requires more time to develop and the likelihood of errors increases. Adding more detail also requires longer computer time to run.

• Analysis of the simulation output: Simulation generates much raw data, which must be analyzed using statistical techniques before any conclusion can be made.

• Similar to measurements, a careful experiment design is essential to keep the simulation cost down.

For communication networks, developing a simulation program requires:

• Modeling user demands for network resources.

• Characterizing how the network resources are assigned for processing user demands.

• Estimating system performance based on output data generated by the simulation.

The stochastic nature of demands for network resources is modeled using pseudo-random number generators. The operation of communication networks is described by simulation programs.

5.2 Discrete-Event Simulation

The kind of simulation of interest to us is discrete event simulation [13], where the state of the system is updated every time an event occurs. From a high-level perspective, telecommunication networks can be seen as consisting of users that generate demands for network resources, and protocols that control the allocation of network resources to satisfy those demands. The generation of user demands and their satisfaction are encapsulated in simulation events, which are ordered by their time of occurrence. The action of the protocols depends on the state of network at the time the demand was issued. This event-based processing lends itself to a method known as discrete-event simulation (DES). Most simulation tools for telecommunications networks are based on DES. An important characteristic of DES models is that they keep time via simulation clocks, which changes by random increments. The basic executable unit in DES models is an event (a program that is executed at discrete simulation times).

In DES, the state of the simulated system is stored in a set of system state variables. Event routines cause state variables to be modified. An event list is used to control the execution sequence of these event routines; the list consists of events in increasing chronological order. Event routines can add or delete items from the event list, and pseudo-random number generators in the event routines provide the requisite randomness for modifying and scheduling of future events.

Typical events in a communications network simulation include the arrival of demands for network resources. A description of network resources, which is needed to satisfy the demand, is associated with each arrival.

Traffic modeling is a key element in simulating communications networks. A clear understanding of the nature of traffic in the target system and subsequent selection of an appropriate random traffic model are critical to the success of the modeling enterprise.

5.3 System Model and Definitions

5.3.1 Fundamentals of Queuing Model

In cellular communication networks, many users share the network resources such as radio spectrum. Since the radio spectrum is limited, some call requests may be denied. For channel assignment algorithms, the most important performance metric is the probability of a call request being blocked (*blocking probability*). For a given algorithm, the blocking probability is solely decided by two factors: how frequently the call requests are being made and for how long the calls will stay using the radio channels.

The Poisson Process

The *Poisson* model is the most commonly used traffic model. A *Poisson process* can be characterized in terms of the time interval between arrivals. If the call requests occur at times T_1, T_2, ..., T_{i-1}, T_i, the inter-arrival times $\tau_i = T_i - T_{i-1}$ are identically distributed random variables and are exponentially distributed with rate parameter λ [9]:

$$P\{\tau \leq t\} = 1 - \mathrm{P}\{\text{no call request in } t \text{ seconds}\} = 1 - e^{-\lambda t}$$

When we look at a very small time interval $(t, t + \delta)$, we have another definition of the Poisson process [9]:

$$P_0(\delta) = \mathrm{P}\{\text{no call request in } (t, t + \delta)\} = 1 - \lambda\delta$$

$$P_1(\delta) = \mathrm{P}\{\text{exactly one call request in } (t, t + \delta)\} = \lambda\delta$$

$$P_2(\delta) = \mathrm{P}\{\text{more than one call request in } (t, t+\delta)\} = 0$$

Equivalently, a Poisson process can be characterized by a counting process: the average number of call requests in t seconds is λt and the probability of k call requests is given by:

$$\mathrm{P}(k \text{ call requests in } t \text{ seconds}) = \frac{e^{-\lambda t}(\lambda t)^k}{k!}$$

and the numbers of call requests in disjoint intervals are independent random variables [9].

The call durations are modeled by an exponential process with a mean $1/\mu$. The probability of a call ending during time interval $(T, T+\delta)$ is $\mu\delta$. We define the *system load* as the average number of call arrivals divided by the maximum capacity of the system.

Birth-Death Process

For a *birth-death process*, the state of a system can be represented by the number of ongoing calls in the system, say n. Assume in an incremental time interval at most one call arrival or call departure is possible. An arrival of a new call causes that number to be changed to $n+1$, which is called a *birth*. Similarly, the termination of a call changes the state of system to $n-1$, which is called a *death*.
The state transition flow diagram for a birth-death process is shown in Figure 5.1. When the system is in state n, the rate of new arrivals is λ_n, the rate of departure is μ_n. Let P_n be *steady-state probability* of a birth-death process [13]:

$$P_n = \mathrm{Prob}(\text{Number of ongoing calls in the system} = n) \qquad n = 0, 2, \cdots N \quad (5.1)$$

In particular, P_0 represents the probability of that the system has no call going on. P_n is the probability of a new coming call being blocked as the system is already full.

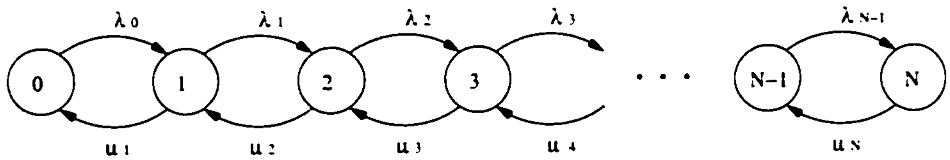

Figure 5.1: General state transition flow diagram for a birth-death process

The probability distribution of a steady state birth-death is:

$$P_n = \frac{\lambda_0\lambda_1\cdots\lambda_{n-1}}{\mu_1\mu_2\cdots\mu_n}P_0, \qquad n = 1, 2, \cdots N \tag{5.2}$$

where P_0 can be obtained using the requirement that all probabilities sum to 1 [13]:

$$P_0 = [1 + \sum_{n=1}^{n=N} P_n]^{-1} \tag{5.3}$$

5.3.2 Simulation Environment

The simulated cellular network has $7 \times 7 = 49$ cells as shown in Figure 5.2. The cell location is identified by an integer variable $i (0 \le i \le 48)$. The 7×7 network has been used as in [12, 27] because it is big enough to show many details while small enough to handle. A reuse distance of two is assumed. This implies that a channel should not be used for more than one user at the same time in the same cell or in immediate neighboring cells. For example, in Figure 5.2, if a channel is being used in cell 24, it should not be used concurrently by another user in cells 16, 17, 23, 24, 25, 31 and 32.

We choose to evaluate the channel assignment algorithms' performance under two environments, *uniform* and *non-uniform* network traffic distribution. In the first model, we assume each cell has the same traffic load,*i.e.* call arrival rate to a base station is equal to λ, and the average call duration is $1/\mu$. However, in a *non-uniform* model, the simulated cellular network has four isolated hot-spot cells 8,18 29 and 39. In the hot-spot cells, the traffic loads are higher than that of normal cells. To investigate the algorithms' adaptabilities to traffic variability, several scenarios, where

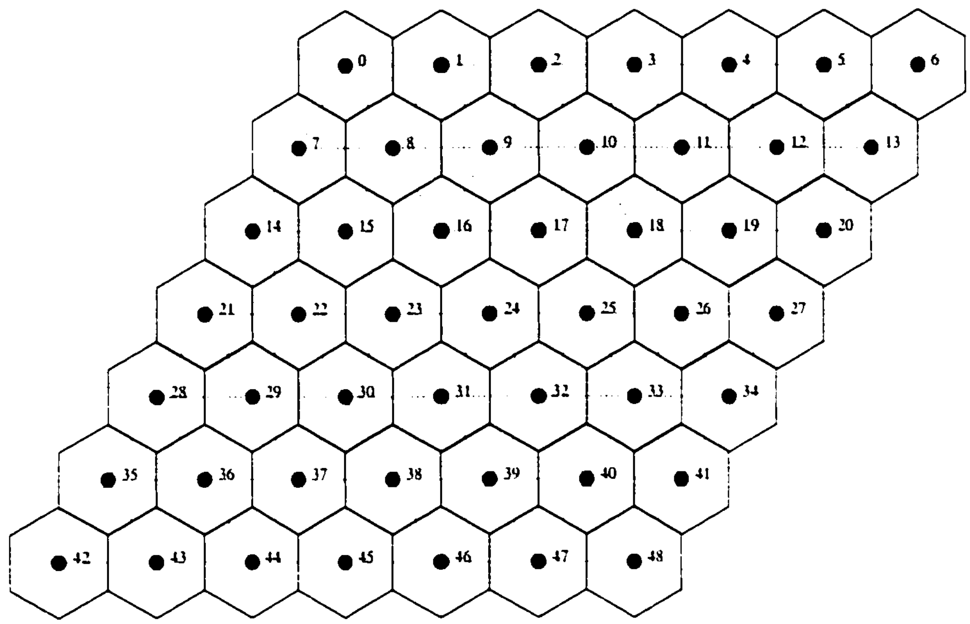

Figure 5.2: A 7 × 7 grid cellular system.

the traffic loads in hot-spot cells are set to be 2. 3 or 4 times that in normal cells were simulated.

In order to study the performance of channel assignment algorithms. we have designed a discrete-state model to simulate the cellular network. All the radio channels in the network are represented by a set of positive integers. Call events are generated by a random number generator and stored in an ordered event list. Each call event is represented by a event record containing the time and the place at which the event should occur. Each base station has a channel occupancy list which contains the information about which call sessions are going on within the cell and which channels are being used. When a new call event is generated in a cell, the base station will search for an available channel according to the channel assignment algorithm. When a channel is found to be available for the new call, it will be assigned to the new call and the channel occupancy list of the cell will be updated, then go to the next earliest occurring event. The ongoing calls always have higher priorities than newly generated calls and will never be dropped or reassigned other channels to make room

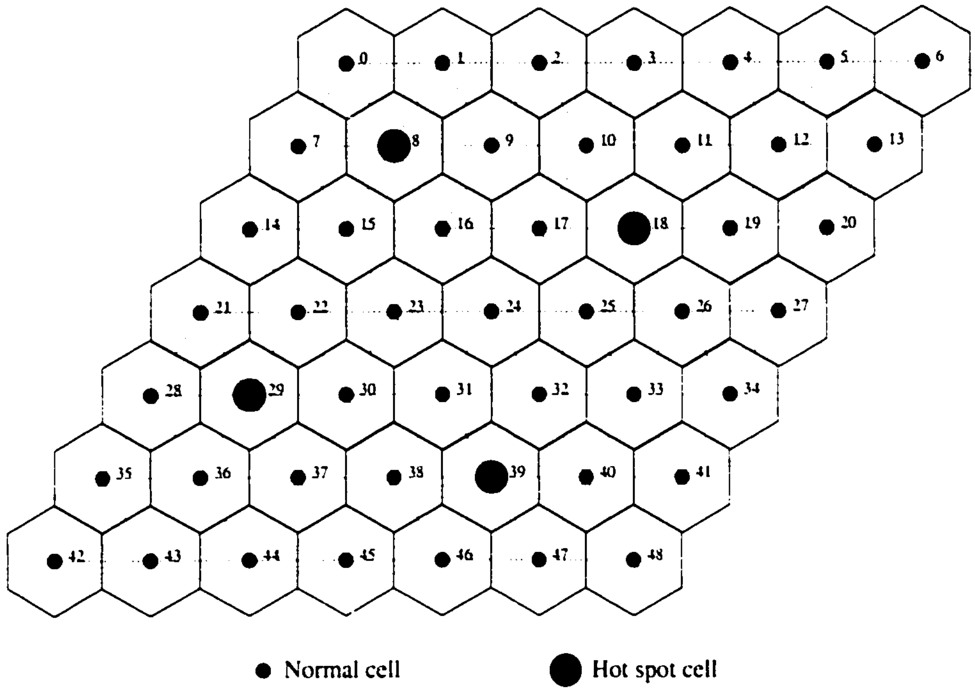

Figure 5.3: A 7×7 grid cellular system with 4 hot-spot cells.

for new calls. If there is no channel available for the new call, the call is blocked and will never return to the system. In our experiments, we have set the number of channels to be 30.

5.3.3 Performance Metrics

In the literature, various attributes of the channel assignment algorithms have been studied. These include the blocking probability [1, 18, 12, 19, 27, 3, 35], achievable system capacity [24], and number of messages [1, 27].

In this section, we define the performance evaluation metrics used in our simulation. In particular, the blocking probability, system capacity, the fairness index and the channel localization index were considered.

Blocking Probability

Two parameters of importance are *system load* and *blocking probability*. Let λ_i be the arrival rate and $1/\mu_i$ be the mean call duration in cell C_i. According to Little's law, the average number of channels needed to support all arrived calls in a cell is λ/μ. Assume that the total number of radio channels in the system is N. The mean number of channels allocated to each cell and therefore, the maximum mean number of concurrent calls in a cell, is $N/3$. Assume there are M cells $C_1, C_2, \ldots, C_M$ in the system. We define the *system load* as:

$$System\ load = \frac{\sum_{i=1}^{i=M} \frac{\lambda_i}{\mu_i}}{\frac{N}{3} \times M} \times 100$$

For example, for a 49 cell, 30 channel system, if the call arrival rate is 0.1 call/cell/second and the mean duration of a call is 180 seconds in each cell, the system load is equal to $\frac{49 \times 0.01 \times 180}{\frac{30}{3} \times 49} \times 100 = 18\%$.

The *blocking probability* at load l is defined as

$$Blocking\ probability(l) = \frac{Total\ number\ of\ blocked\ calls\ at\ load\ l}{Total\ number\ of\ call\ requests\ at\ load\ l} \times 100.$$

System Capacity

One objective of cellular architecture is to increase the capacity of the system, *i.e.* how many active or potentially active users the system can support. For any given system, the capacity cannot be increased without compromising the blocking probability. The more users in the system, the more likely a user's call is to be blocked. Blocking probability at a particular load can be used to determine the capacity of a scheme.

Usually cellular systems are required to have a blocking probability no more than

2%: no more than 2% of all call requests shall be blocked even during the busiest hour [4]. The maximum traffic load a system could provide without violating this 2% blocking standard can be used to evaluate the performance of channel assignment algorithm. The formal definition of *system capacity* that we use is given bellow:

$$System\ Capacity = Max\{l \mid Blocking\ probability(l) \leq 2\%\}$$

Fairness Index

Due to the fluctuation in the traffic and the geographic position in the network. the users in different cells may experience different service quality. Fairness in service quality is also of interest to us. We define the *fairness index* as follows:

$$Fairness\ Index = \frac{Maximum\ Blocking\ Probability\ Over\ All\ Cells}{Blocking\ Probability\ Averaged\ Over\ All\ Cells}$$

Note that the fairness index is always at least 1. The higher the fairness index is, the more unfair the algorithm. A fairness index of one means an absolutely fair channel assignment algorithm.

Channel Localization Index

In [17, 26], FCA has been shown to perform better than DCA when system load is high and the reason is due to the trade-off between reusability and flexibility in using channels. In FCA, all the *red* cells will always use the same set of red channels, all the *blue(green)* will always use blue(green) channels. In DCA there is no limitation on which channels to use. Any cell, no matter what the base color it has, can use any channel as long as its not being used by its neighbors. In FPA, while all channels are available to every cell in principle, different cells use channels in different orders.

Therefore we can expect some kind of channel localization in practice. To study this effect, we investigate patterns of locality in channel usage.

Because of the symmetry among different base colors, we investigated the channel usage on *red* cells only and the results are expected to be similar to those on *blue* and *green* cells. We define the *channel localization index(CLI)* of *red* cells as:

$$\textit{Channel Localization Index} = \frac{\textit{Number of Times Red Channels Have Been Used}}{\textit{Number of Times Channels Have Been Used}} \times 100$$

A CLI of 100% means the algorithm never uses channels other than its own set of channels. An algorithm with absolutely no preference on the channel's color should have a CLI of 33.3%.

5.3.4 Analytical Models

Despite considerable efforts [18, 33, 19, 29] in using analytical methods to analyze the performance of some special classes of channel assignment algorithms, in general, it is difficult to calculate the blocking probability directly from the above analytical models for any reasonable size of system because of the large number of states. For example, in a 30 channel, 49 cell system, there are 10^{49} different states and to calculate the probability of each of these states becomes a formidable task. In this thesis we shall construct queuing models of three channel assignment algorithms for a 3 cell cluster as shown in Figure 5.4, and compare the analytical solutions with the simulation results. The close match of the two data sets indicates that the simulation program was working properly.

If the number of channels assigned to each cell is fixed, *i.e.* 10 channels per cell, and every cell is subject to the same traffic pattern, it is enough to focus the performance evaluation on only one cell.

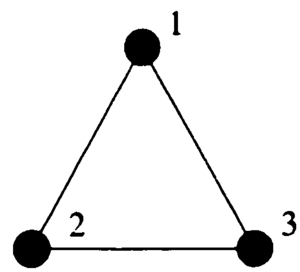

Figure 5.4: A three cell cluster

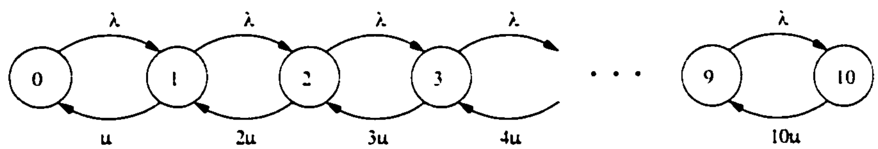

Figure 5.5: State transition flow diagram–fixed channel assignment

The state transition flow diagram for fixed channel assignment is shown in Figure 5.5. New calls are coming in at rate λ as long as the there are less than 10 calls in the system. When the system is full, arriving call requests are lost. The probability of a call ending is proportional to the number of active calls in the system. These can be expressed as:

$$\lambda_n = \begin{cases} \lambda & \text{if } n < 10; \\ 0 & \text{if } n \geq 10. \end{cases}$$

$$\mu_n = \begin{cases} n\mu & \text{if } n \leq 10; \\ 10\mu & \text{if } n > 10. \end{cases}$$

The solution can be obtained by substituting these coefficients into equations (5.2) and (5.3). Figure 5.6 shows the comparison of the analytical results derived from the queuing model and the results of simulation carried out for all 3 cells in the network.

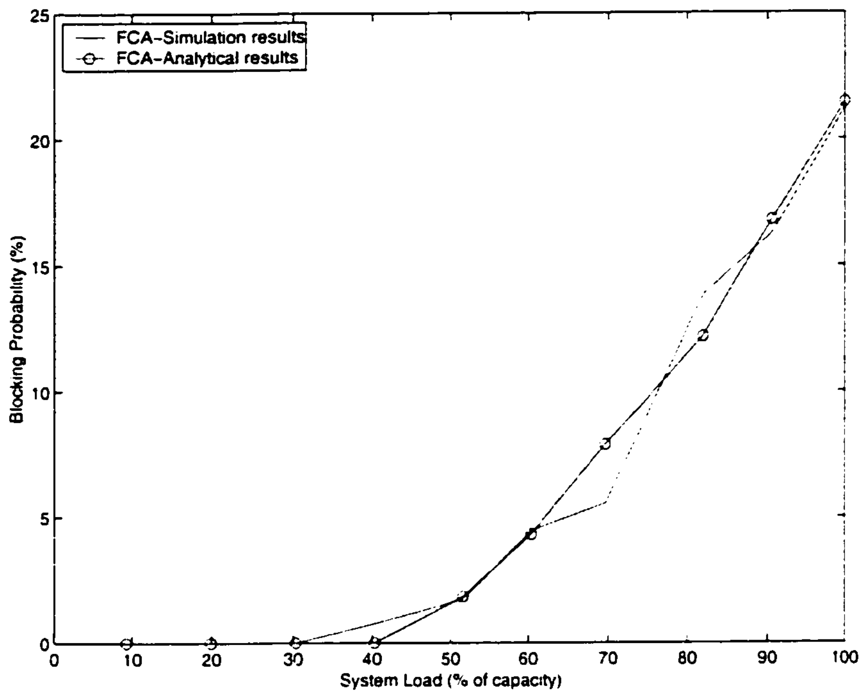

Figure 5.6: Fixed channel assignment:Simulation vs analytical

Dynamic Channel Assignment Algorithm

In the dynamic channel assignment algorithm, all the channels are put into a pool and shared by all three cells. We can look at the three cell cluster as a single big cell, as a call request in any of these three cells will be blocked if and only if all the 30 channels are being used by 30 calls in three cells. Similar to the case of fixed channel assignment algorithm, we can model this system by a single queue in which the coefficients can be described as follows:

$$\lambda_n = \begin{cases} 3\lambda & \text{if } n < 30; \\ 0 & \text{if } n \geq 30. \end{cases}$$

$$\mu_n = \begin{cases} n\mu & \text{if } n < 30; \\ 30\mu & \text{if } n \geq 30. \end{cases}$$

The state transition flow diagram for dynamic channel assignment is shown in

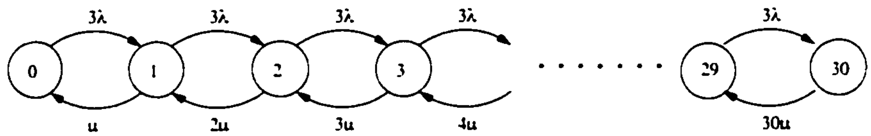

Figure 5.7: State transition flow diagram–dynamic channel assignment

Figure 5.7.

Figure 5.8 shows the comparison of the analytical results derived from the queuing model and the results of simulation carried out for all 3 cells in the network.

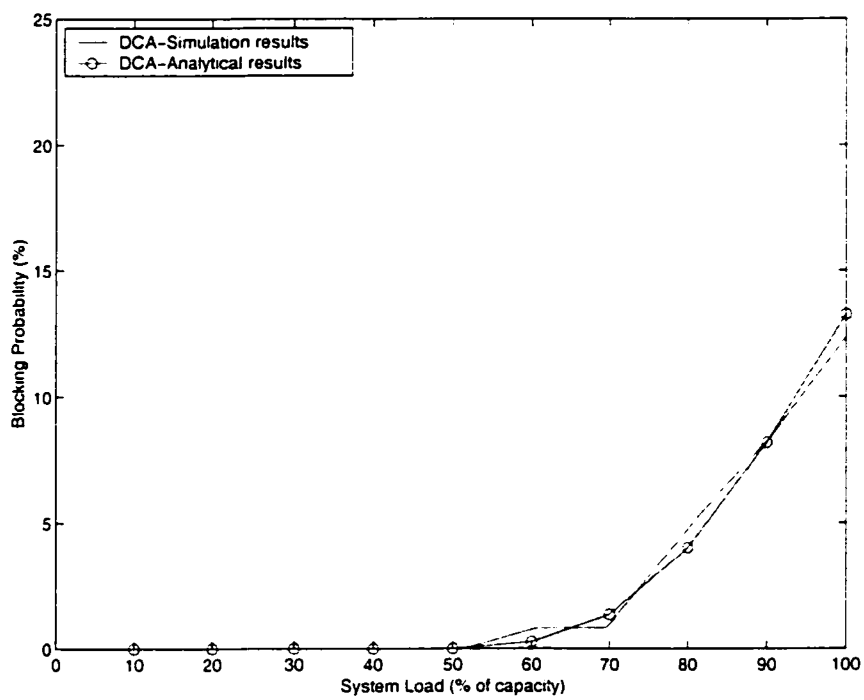

Figure 5.8: Dynamic channel assignment:Simulation vs analytical

Fixed Preference Assignment Algorithm

Let the whole spectrum be divided into 3 subgroups *Red*, *Blue* and *Green*, each consisting of 10 channels. When a new call request comes in, cell 1 searches for available channel in all 3 subgroups following the order of *Red-Blue-Green*, cell 2

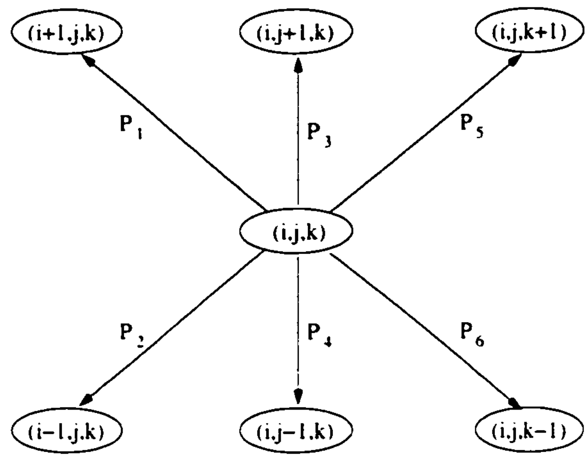

Figure 5.9: State transition flow diagram—fixed preference channel assignment

follows the order of *Blue-Green-Red*, and cell 3 *Green-Red-Blue*.

Let i be the number of red channels being used in all three cells, j be the number of blue channels being used, and k be the number of green channels being used. The state of the system therefore can be represented by a 3-tuple (i, j, k) where $0 \le i, j, k \le 10$. The following coefficients describe the system:

$$P_1 = \begin{cases} \lambda & \text{if } i < 10, j < 10, k < 10; \\ 2\lambda & \text{if } i < 10, j < 10, k = 10; \\ 3\lambda & \text{if } i < 10, j = 10, k = 10; \\ 0 & \text{if } i \ge 10, j \ge 10, k \ge 10; \end{cases} \qquad P_2 = \begin{cases} i\mu & \text{if } i < 10; \\ 10\mu & \text{if } i \ge 10; \end{cases}$$

$$P_3 = \begin{cases} \lambda & \text{if } i < 10, j < 10, k < 10; \\ 2\lambda & \text{if } i = 10, j < 10, k < 10; \\ 3\lambda & \text{if } i = 10, j < 10, k = 10; \\ 0 & \text{if } i \ge 10, j \ge 10, k \ge 10; \end{cases} \qquad P_4 = \begin{cases} j\mu & \text{if } j < 10; \\ 10\mu & \text{if } j \ge 10; \end{cases}$$

$$P_5 = \begin{cases} \lambda & \text{if } i < 10, j < 10, k < 10; \\ 2\lambda & \text{if } i < 10, j = 10, k < 10; \\ 3\lambda & \text{if } i = 10, j = 10, k < 10; \\ 0 & \text{if } i \geq 10, j \geq 10, k \geq 10; \end{cases} \qquad P_6 = \begin{cases} k\mu & \text{if } k < 10; \\ 10\mu & \text{if } k \geq 10; \end{cases}$$

Figure 5.10 shows the comparison of the analytical results derived from the queuing model and the results of simulation carried out for all 3 cells in the network.

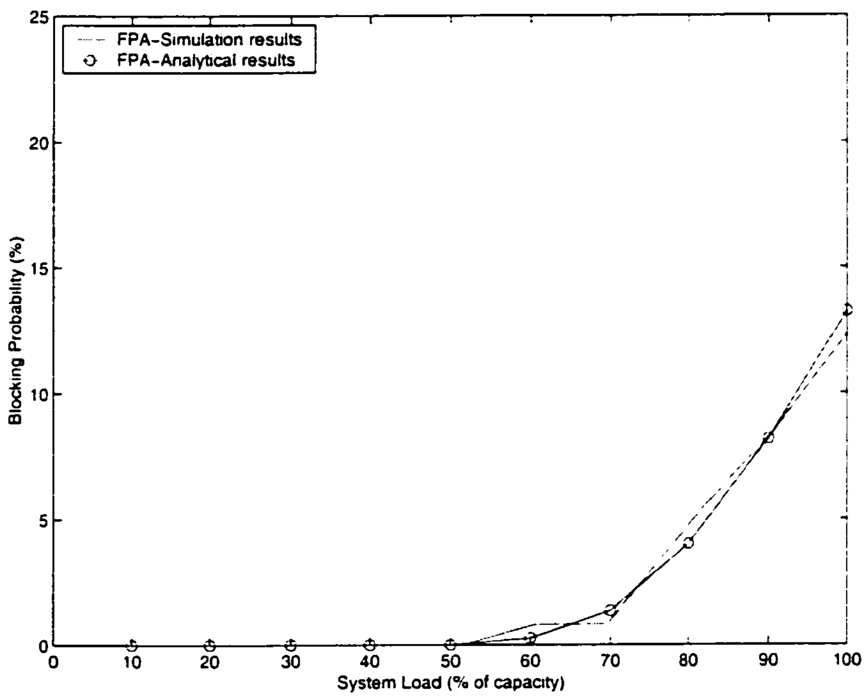

Figure 5.10: Fixed preference channel assignment:Simulation vs analytical

As we can see from Figure 5.8 and Figure ??, in a 3-cell system, dynamic and fixed preference channel assignment algorithms have exactly the same blocking probability and both of them are better than fixed channel assignment algorithm. In fact, dynamic and fixed preference channel assignment algorithms are the optimal algorithms because a call request is blocked if and only if all the channels are in use and no other algorithm could do better in this situation.

5.4 Experimental Results

5.4.1 Uniform Traffic

The comparison of performance of the three algorithms under uniform traffic in a 30 channel system has been done for system load increased from 10% to 100%. The simulation is carried out for the 49 cell network and the results have been shown in Figure 5.11.

Blocking Probability

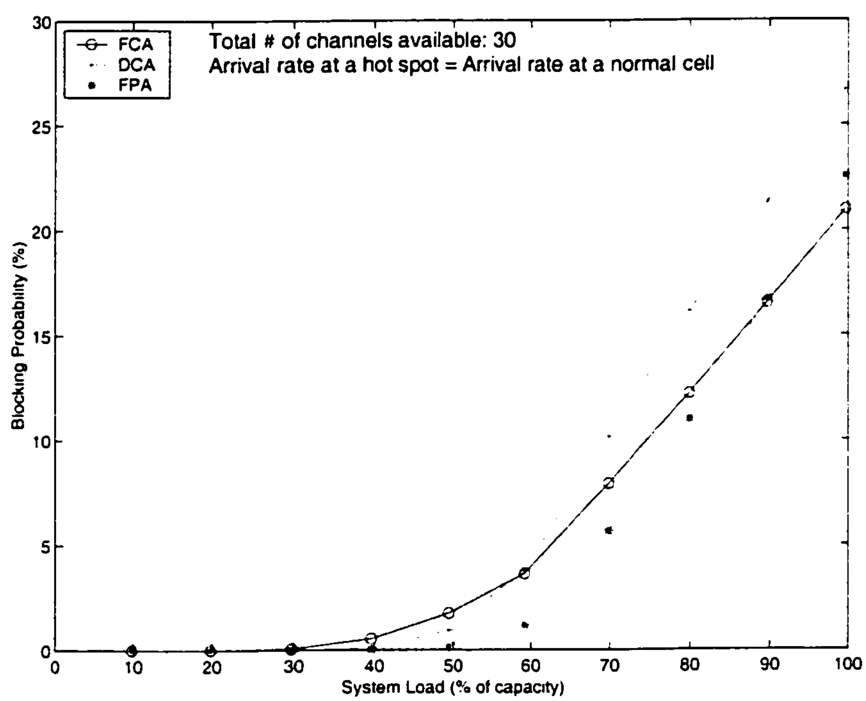

Figure 5.11: Blocking probability vs system load using uniform traffic model

As shown in Figure 5.11, all the algorithms have blocking probability of 0 until the system load exceeds 30%. The FPA algorithm has the lowest blocking probability among the three algorithms until system load reaches a very high level of 90%. When system load is above 90%. FCA has the lowest blocking probability while no algorithm can achieve a blocking probability below 16%, which is considered to be too high for phone services. Despite seeming to have the greatest flexibility in finding available channels it has, the dynamic algorithm has only moderate performance when the system load is below 55%, at which point it has a blocking probability of 3% and after that it starts having higher blocking probability than other two algorithms by 2% to 5%.

System Capacity

As shown in Figure 5.11, at a blocking requirement of 2%, the FPA algorithm provides the highest system load of about 61%. FCA and DCA have almost the same maximum system load, which is about 10% lower than that of FPA.

Fairness

As we can see from Figure 5.12, under the uniform traffic pattern, FCA provides the best fairness performance. This is of no surprise because every cell has the same system load and same number of channels, the blocking probabilities are to be similar over all cells in the system. The fairness performance of DCA and FPA are very poor until the system load reaches 70%. When system load level is above 70%, there is very little difference among the three strategies.

Local Channel Usage

As we can see from Figure 5.13, the *channel localization index (CLI)* of FCA is always 100% as expected. The CLI of DCA drops from the initial 100% level rapidly as the system load increases and finally stabilizes at about 40% level when system load

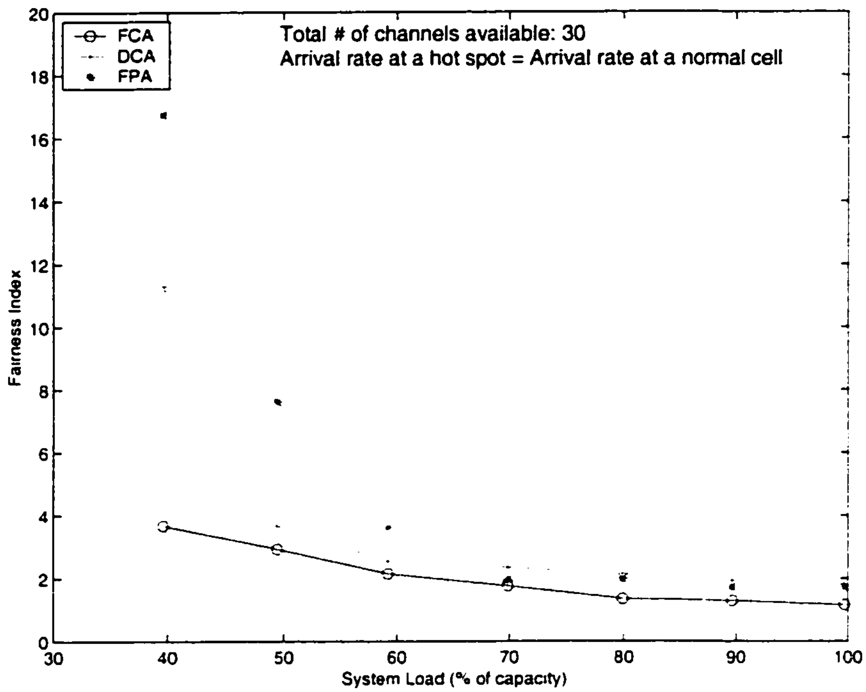

Figure 5.12: Fairness index:Uniform traffic

exceeds 50%. The fact that DCA's CLI is 40% shows that DCA is very close to the expected behavior of an algorithm showing no preference between the channel sets, which would have a CLI of 33%. A somewhat surprising observation is that the FPA has shown a very strong locality in the use of channels. Despite its ability of using any channel, FPA has a channel usage pattern that is very similar to FCA, in fact, more than 98% of all channels it used are from its own set until the system load is over 50%. After that, FPA still uses more than 80% from its own set of channels even when the system load is approaching an extremely high level of 100%.

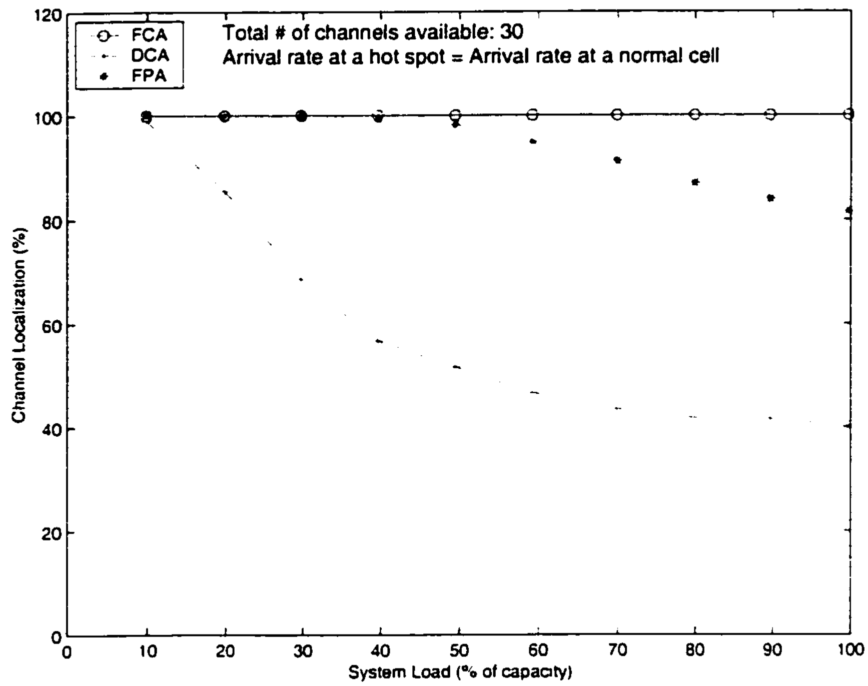

Figure 5.13: Local channel usage

5.4.2 Non-Uniform Traffic

Blocking Probability

Under the non-uniform traffic pattern, while all the algorithms have higher blocking probabilities, the existence of hot-spot cells has much more negative impact on the FCA algorithm than other algorithms. The difference between FCA and other algorithms becomes more obvious when the hot-spot cells have higher load compared to normal cells. As we can see from Figures 5.14, 5.15, and 5.16, under the non-uniform traffic pattern, the blocking probability of the three algorithms changes in a similar manner as it does under the uniform traffic pattern. At low system load, FPA has the lowest blocking probability while FCA's blocking probability is the highest. As the system load increases, DCA's blocking probability goes up at a higher speed than the other two algorithms. However, despite the initial higher blocking probability, the

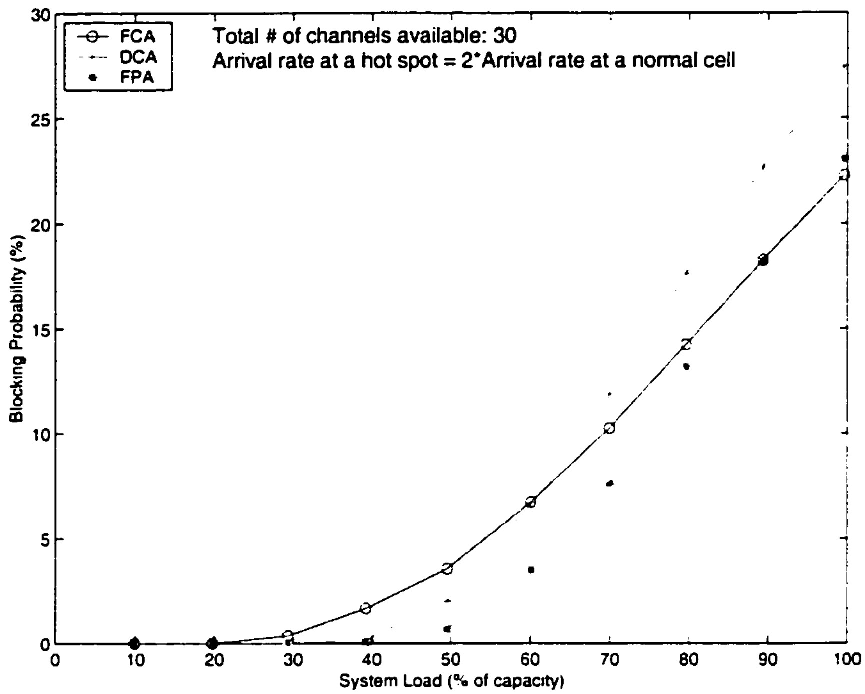

Figure 5.14: Blocking probability vs system load using the non-uniform traffic model

increase of FCA's blocking probability is slower than other two algorithms and finally becomes superior to other two algorithms. Although FCA will gradually outperform DCA and FPA as the system load goes higher no matter how uneven the traffic is. the speed of this changing process is greatly hampered by the variation in the traffic. While FCA's blocking probability is the least sensitive to the increase of system load. DCA and FPA have shown a better adaptability to the fluctuation in the traffic. In particular, when hot-spot traffic is 4 times normal traffic load, FPA has the lowest blocking probability for all loads.

System Capacity

When the blocking probability is set to be 2%, under the non-uniform traffic pattern. the system capacity of FCA is much lower than other algorithms, especially compared to FPA, the capacity of FCA is only about half of that of FPA. The comparison of

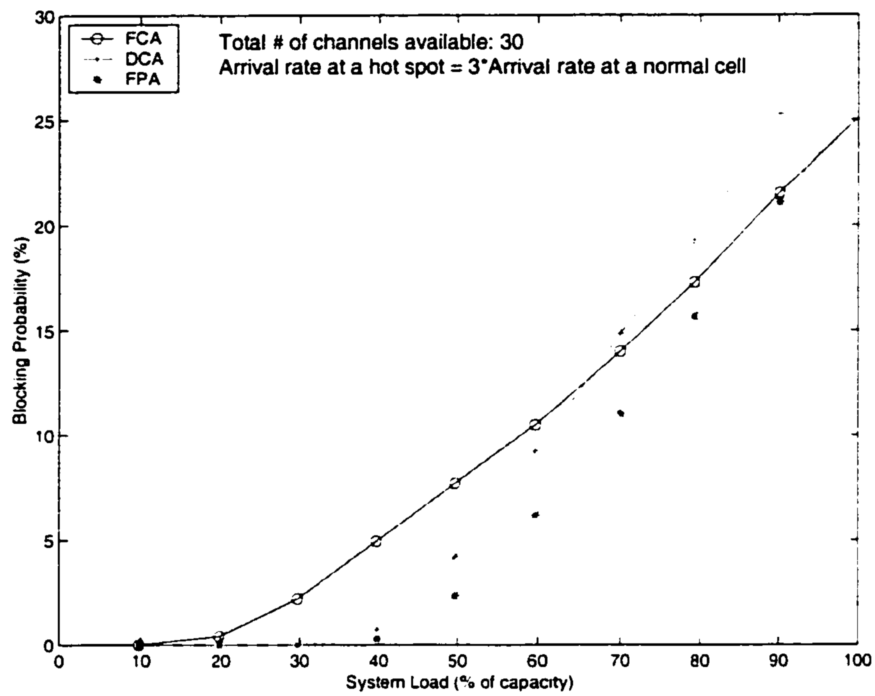

Figure 5.15: Blocking probability vs system load using the non-uniform traffic model

system capacity under different traffic patterns is summarized in Table 5.1:

As we can see from Table 5.1, the system capacity of FCA drops dramatically under the non-uniform traffic. When the load on a hot-spot cell is 4 times the load on a normal cell, the system capacity of FCA drops for more than 30% from that under the uniform traffic, which is much bigger than drops in DCA (13%) and FPA (19%). If we want to keep the blocking probability at a relative low level (2%), FCA no longer has the advantage it enjoys at high load level. FCA has the lowest system capacity and the poorest adaptability to the fluctuation in the traffic. Although FPA has always achieved the highest system capacity in our experiments, DCA's system capacity decreases the least with fluctuation in the traffic.

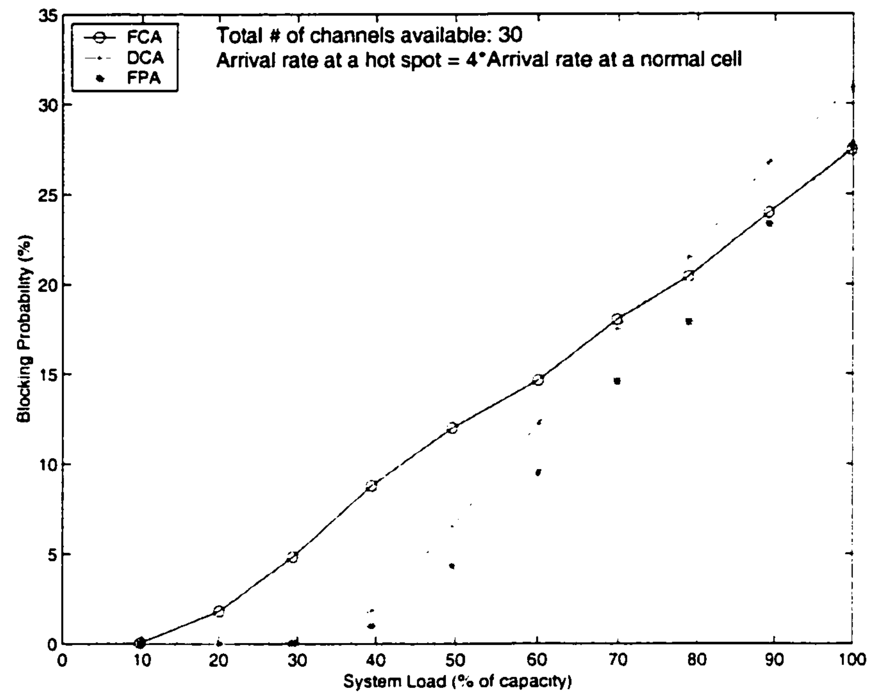

Figure 5.16: Blocking probability vs system load using the non-uniform traffic model

Fairness

As we can see from Figure 5.17, 5.18, and 5.19, unlike under the uniform traffic, the performance of FCA is no longer better than other algorithms when the traffic is not uniform. Especially when the system load is higher than 50%, the fairness index of FCA is higher than DCA by about 20 – 40%. The poor performance of FCA is due to the fact that every cell gets the same number of channels and therefore the blocking probability in hot-spot cells is much higher than that in normally loaded cells. In DCA, a hot-spot cell gets more channels than any of its normal cell neighbors because it has more calls competing for available channels. The flexibility in assigning channels according to demands has contributed to the lowest fairness index of DCA.

Traffic Pattern	System Capacity		
Normal : Hot-spot	FCA	DCA	FPA
1 : 1	51%	53%	61%
1 : 2	41%	50%	60%
1 : 3	29%	43%	48%
1 : 4	20%	40%	42%

Table 5.1: System capacity under different traffic patterns

Local Channel Usage

As we can see from Figure 5.20. Figure 5.21. and Figure 5.22. the CLI of channel assignment algorithms is not affected much by the fluctuation in the traffic distribution with one exception of FPA. Although the same as in the even traffic distribution scenario. in terms of always using more than 80% from its own set of channels. FPA starts to use channels outside its own set earlier. This can be explained as follows: The higher demand in the hot spot cells makes such cells look for channels in its neighborhood. *i.e.* outside its primary channel set. more frequently. When traffic is low in the neighboring cells. this "borrowing" is more successful. However. when the traffic load is high everywhere, a cell looking for a channel outside its primary set is much less likely to be successful. In this way. a cell is "forced" to use only its primary channels.

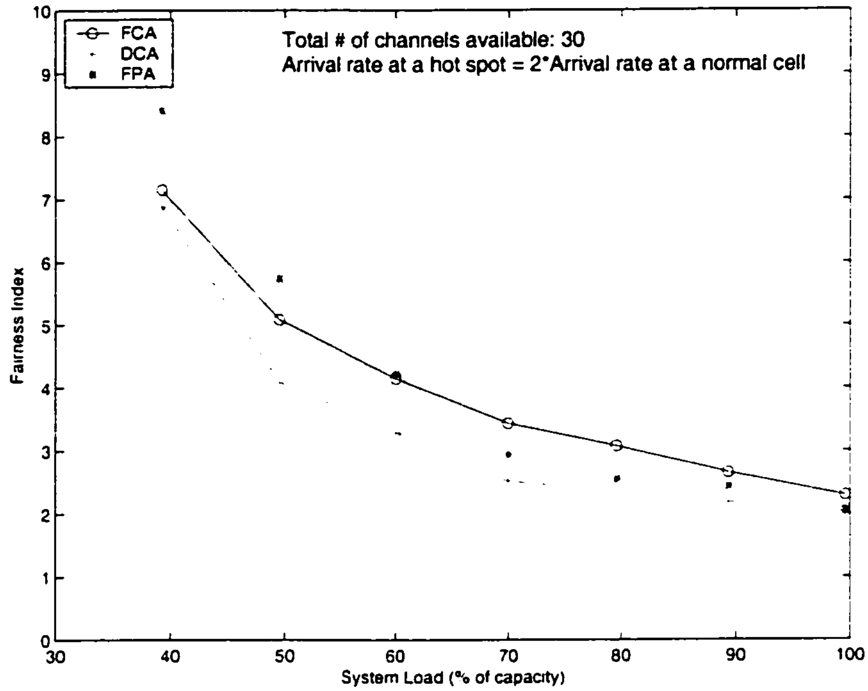

Figure 5.17: Fairness index:non-uniform traffic

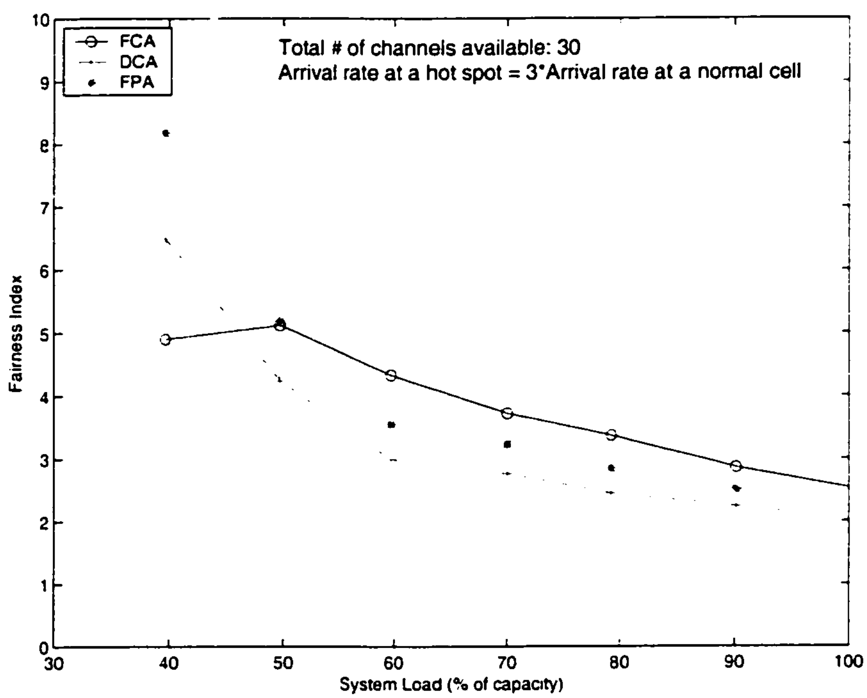

Figure 5.18: Fairness index:non-uniform traffic

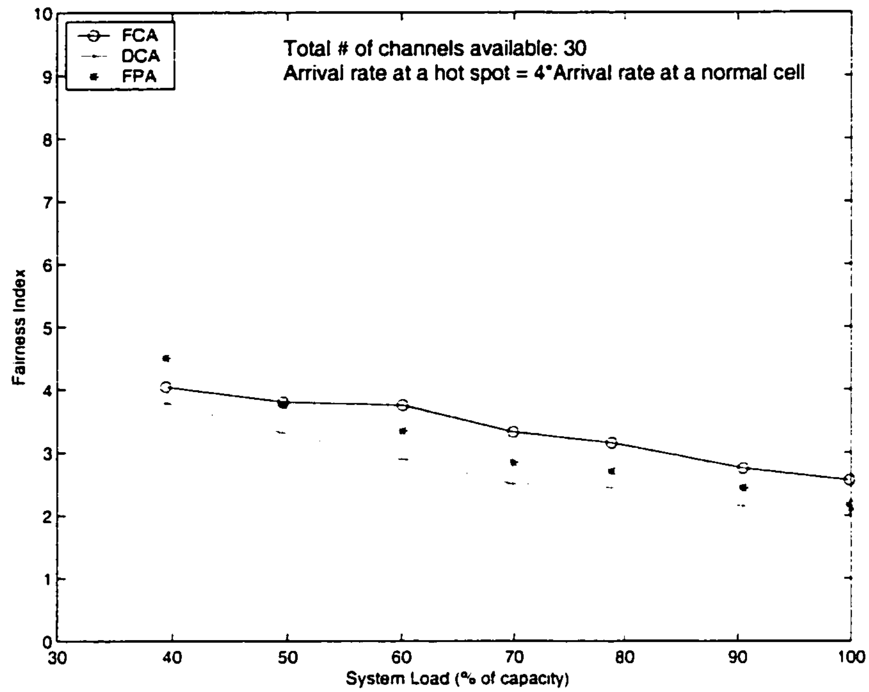

Figure 5.19: Fairness index:non-uniform traffic

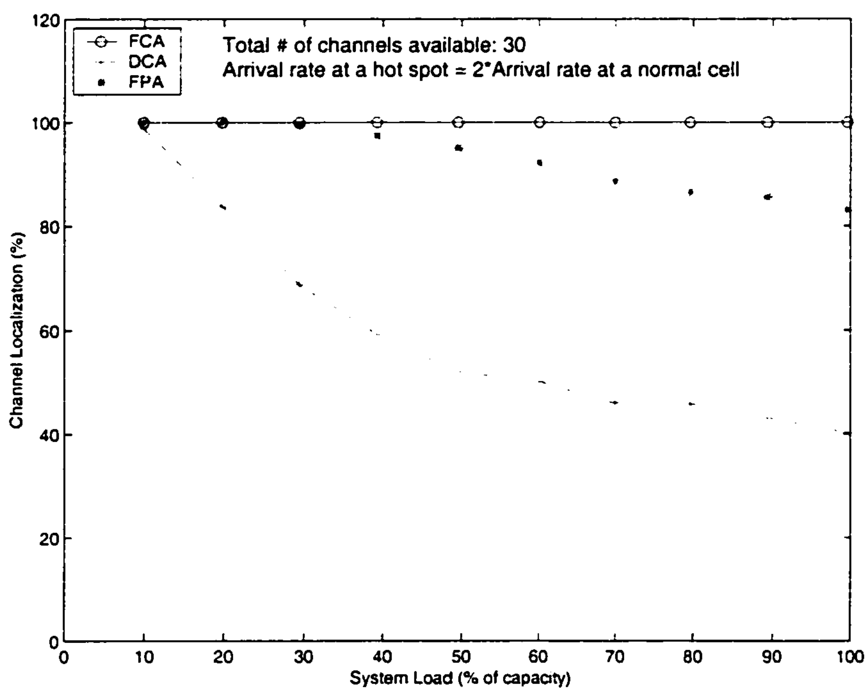

Figure 5.20: Local channel usage

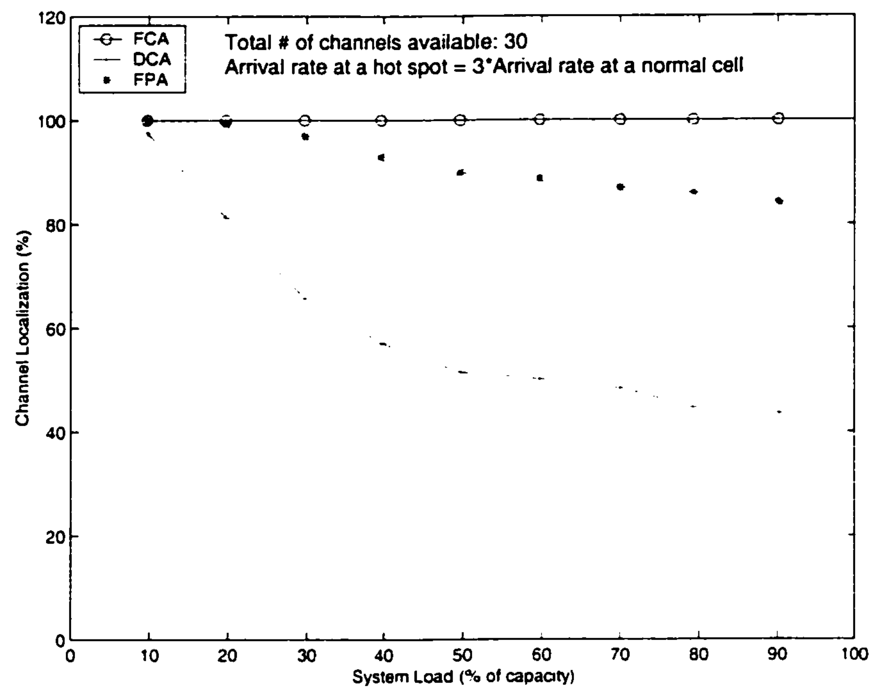

Figure 5.21: Local channel usage

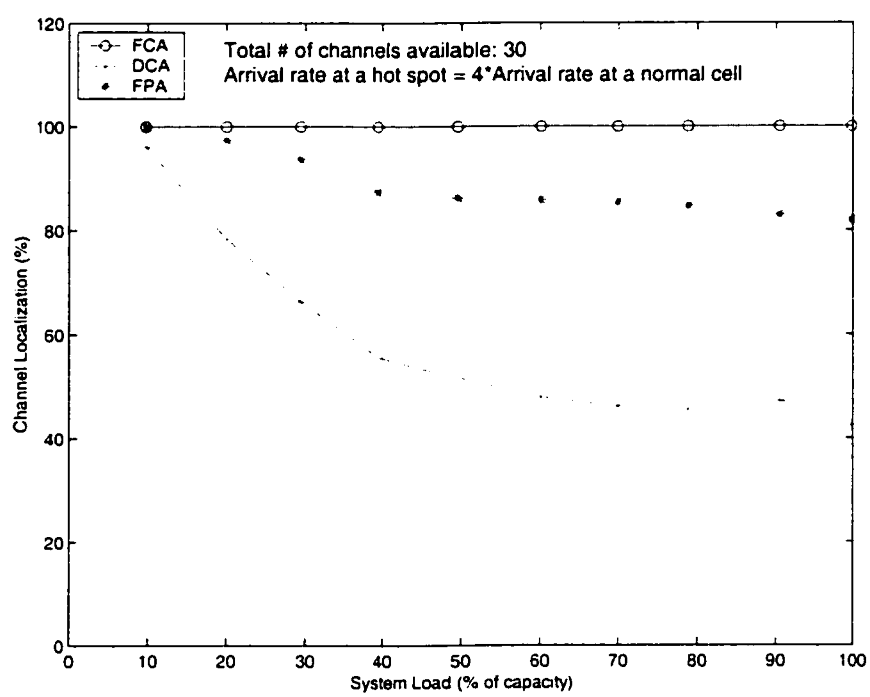

Figure 5.22: Local channel usage

Chapter 6

Conclusions and Future Work

In this thesis, we have shown upper and lower bounds for the competitive ratio of a widely studied version of dynamic channel assignment, which we call the greedy algorithm, for arbitrary reuse distance r. We studied both offline and online versions of this algorithm. No channel reassignments are allowed in the online version. We also proposed a new online algorithm for channel assignment, based on the concept of preference lists, and proved a lower bound on its competitive ratio.

Our results show that for the offline case, the greedy algorithm is provably outperformed by borrowing strategies for both reuse distance 2 and 3. For the online case, there is a dearth of algorithms with proven bounds on the competitive ratio. While we showed bounds on the competitive ratio of the greedy algorithm without channel reassignments for both reuse distance 2 and 3, these bounds are not tight. Our results show that the greedy algorithm is at least as good as any other known online algorithm without reassignments for reuse distance 2 and 3. Our results also showed the fixed preference assignment algorithm to be a promising channel assignment algorithm that does not perform reassignments.

It would be interesting to prove exact bounds on the competitive ratio of the online greedy algorithm without reassignment of channels for all values of reuse distance, particularly $r = 2$ and 3. Finding other online algorithms that do not perform reassignment is also an urgent area of research.

Our simulation experiments show that among three channel assignment algorithms we studied, FA has the lowest blocking probability under heavy traffic load but the highest under light traffic. FPA has a very low blocking probability except at a very high level, while the greedy DCA algorithm never has the lowest blocking probability for any traffic load. In terms of maximum system capacity at an acceptable level of blocking probability, FPA has a great advantage over other two algorithms, even under extremely uneven traffic distributions. FCA has the lowest system capacity, especially when the traffic distribution is uneven. The system capacity FCA provides can be as low as less that half of that of FPA. While DCA provides moderate system capacity, its advantage is that it has the best adaptability to variations in the traffic load.

FCA's fairness index is the best only under the unfair traffic pattern, which is to be expected. DCA has the best fairness index, especially with uneven traffic and at high loads. The difference between DCA's fairness index and other two algorithms is not as big as expected and remains to be improved. One possible solution is to incorporate a blocking probability feedback mechanism. When a cell experiences a high blocking probability, it can be seen as an indication of heavy traffic in the cell. In order to get more channels to meet the demands, the heavy loaded cell will become more aggressive in acquiring channels, which means the cell could reserve some unused channels for future possible use. By the means of reservation, the hot-spot cells are able to obtain more channels to use and therefore decrease their blocking probability at the price of lower under-utilization of channels and increase in blocking probability within surrounding moderately loaded cells.

The simulation results show that despite the ability to use any channel as in DCA, FPA has shown a very strong locality in channel use. Finding an analytical modeling of this phenomenon would definitely help toward a theoretical analysis of FPA.

Bibliography

[1] S. Hong A. Boukerche and T. Jacob. A performance study of a distributed algorithm for dynamic channel allocation. In *Proceedings of the 3rd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, pages 36–43, 2000.

[2] J. A. Bondy and U. S. R. Murty. *Graph Theory with Applications*. Macmillan Press, 1976.

[3] S. Borst and P. Whiting. Achievable performance of dynamic channel assignment schemes under varying reuse constraints. *IEEE Transactions on Vehicular Technology*, 49(4):1248–1263, 2000.

[4] G. Calhoun. *Digital Cellular Radio*. Artech House, 1988.

[5] I. Caragiannis, C. Kaklamanis, and E. Papaionnou. Efficient online communication in cellular networks. In *Symposium on Parallel Algorithms and Architecture*, pages 46–53, 2000.

[6] D. C. Cox and D. O. Reudink. Dynamic channel assignment in two dimension large-scale mobile radio systems. *Bell Sys. Tech. J.*, 51:1611–28, 1972.

[7] Tomás Feder and Sunil M. Shende. Online channel allocation in FDMA networks with reuse constraints. *Inform. Process. Lett.*, 67(6):295–302, 1998.

[8] M. Frodigh. Reuse partitioning combined with traffic adaptive channel assignment for highway microcellular systems. In *Globecom'92*, pages 1414–1418, 1992.

[9] J. F. Hayes. *Modeling and Analysis of Computer Communications Networks*. Plenum Pub Corp, 1984.

[10] D. Hong and S. Rappaport. Traffic modelling and performance analysis for cellular mobile radio telephone systems with prioritized and nonprioritized handoffs procedures. *IEEE Transactions on Vehicular Technology*, 35(3):77–92, 1986.

[11] C. L. I and P. H. Chao. Local packing-distributed dynamic channel allocation at cellular base station. In *Globecom'93*, pages 293–300, 1993.

[12] K. Kilakos J. Janssen and O. Marcotte. Fixed preference channel assignment for cellular telephone systems. *IEEE Transactions on Vehicular Technology*, 48(2):533–541, 1999.

[13] Raj Jain. *The Art of Computer Systems Performance Analysis*. John Wiley & Sons, 1991.

[14] J. Janssen, D. Krizanc, L. Narayanan, and S. Shende. Distributed online frequency assignment in cellular networks. *Journal of Algorithms*, 36:119–151, 2000.

[15] S. Jordan and A. Khan. A performance bound on dynamic channel allocation in cellular systems: Equal load. *IEEE Transactions on Vehicular Technology*, 43(2):333–344, 1994.

[16] S. Jordan and E. J. Schwabe. Worst-case performance of cellular channel assignment policies. *Wireless Networks*, 2:265–275, 1996.

[17] T. Kahwa and N. Georganas. A hybrid channel assignment scheme in large-scale cellular-structured mobile communication systems. *IEEE Transactions on Communications*, 4:432–438, 1978.

[18] F. P. Kelly. Stochastic models of computer communication systems. *Journal of the Royal Statistical Society, Series B*, 47(3):379–395, 1985.

[19] G. J. Foschini L. J. Cimini and Z. Miljanic C. L. I. Call blocking performance of distributed algorithms for dynamic channel allocation in microcells. *IEEE Transactions on Communications*, 42(8):2600–2607, 1994.

[20] W. C. Y. Lee. New cellular schemes for spectral efficiency. *IEEE Transactions on Vehicular Technology*, 36:188–192, 1987.

[21] V. H. MacDonald. Advanced mobile phone service: The cellular concept. *Bell Systems Technical Journal*, 58(1), 1979.

[22] C. McDiarmid and B. Reed. Channel assignment and weighted colouring. Submitted for publication, 1997.

[23] C. McDiarmid and B. Reed. Colouring proximity graphs in the plane. *Discrete Mathematics*, 199:123–137, 1999.

[24] R. J. McEliece and K. N. Sivarajan. Performance limits for channelized cellular telephone systems. *IEEE Transactions on Information Theory*, 40(1):21–34, 1994.

[25] L. Narayanan and S. Shende. Static frequency assignment in cellular networks. *Algorithmica*, 29:396–409, 2001.

[26] K. Okada and F. Kubota. On dynamic channel assignment strategies in cellular mobile radio systems. *IEICE Transactions on Fundamentals*, E75-A(12):1634-1640, 1992.

[27] R. Prakash, N. Shivaratri, and M. Singhal. Distributed dynamic channel allocation for mobile computing. In *Principles of Distributed Computing*, pages 47-56, 1995.

[28] J. B. Punt and D. Sparreboom. Mathematical models for the analysis of dynamic channel selection or indoor mobile wireless communications systems. *PIMRC*, E6.5:1081-5, 1994.

[29] P-A Raymond. Performance analysis of cellular networks. *IEEE Transactions on Communications*, 39(12):1787-1793, 1991.

[30] M. Serizawa and D. Goodman. Instability and deadlock of distributed dynamic channel allocation. *Proc. 43rd IEEE VTC*, pages 528-31, 1993.

[31] S. Tekinay and B. Jabbari. Handover and channel assignment in mobile cellular networks. *IEEE Communication Magazine*, 29, 1991.

[32] van den Heuvel, J., R. Leese, and Shepherd, M. A. Graph labeling and radio channel assignment. *Journal of Graph Theory*, pages 263-283, 1999.

[33] K. L. Yeung and T. P. Yum. Phantom cell analysis of dynamic channel assignment in cellular mobile systems. *IEEE Transactions on Vehicular Technology*, 47(1):190-195, 1998.

[34] M. Zhang. Comparisions of channel assignment strtegies in cellular mobile telephone systems. *IEEE Transactions on Vehicular Technology*, 38:211-215, 1989.

[35] M. Zhang and T. P. Yum. The nonuniform compact pattern allocation algorithm for cellular mobile systems. *IEEE Journal on Vehicular Technology*, 40(2):387-391, 1991.