Goal Programming Approach for Channel Assignment Formulation and Schemes

NG Cho Yiu

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Master of Philosophy
in
Information Engineering

Supervised by

Prof. LOK Tat Ming

Abstract of thesis entitled:

Goal Programming Approach for Channel Assignment Formulation and Schemes

Submitted by NG Cho Yiu

for the degree of Master of Philosophy

at The Chinese University of Hong Kong in July 2005

In this thesis, a goal programming model is proposed for general downlink channel assignment scheme. Unlike many current channel assignment schemes with quality of service (QoS) requirements modeled as system constraints, QoS requirements are formulated as mathematical functions, which are called *unsatisfactory functions*, in the objective function of our schemes. With this formulation, when there is insufficient amount of resource, the proposed schemes can provide the compromise solutions more conveniently without explicit admission controls. Moreover, QoS requirements can be modeled in a more flexible and detailed manner. The system can substitute any QoS function for each user based on his/her application. In this formulation, there are no assumptions about the underlying multiple access scheme except the orthogonality of the logical channels.

Since it is proved to be an NP-hard problem, an iterative algorithm and a greedy algorithm are proposed to provide near-optimal solutions. In addition, two special cases of this model are studied. For these cases, it is proved that optimal solutions can be obtained by polynomial-time algorithms. Simulation results show that with the proposed algorithms, less channel resource is required to meet the client demand.

摘要

這篇論文爲下行線路通道分配問題（downlink channel assignment）提出一個目標規劃（goal programming）模型。許多現今考慮服務質量（QoS）的通道分配策略，把用戶的服務質量要求表達爲模型中的約束條件（system constraint）。不同於這些策略，我把這些服務質量要求模擬成一個名爲「非滿意函數」（unsatisfactory function）的函數，並放於模型中的目標函數（objective function）中。即使當系統沒有足夠的資源，根據我所提出的模型和相應的策略，我們可以更方便地而又不需要利用額外的接納控制（admission control）得出一個妥協解答（compromise solution）。除此之外，服務質量要求也可以在這個模型中有一個更靈活和細緻的表達。在這個模型中，系統可以根據每個用戶的應用程式代入適合的非滿意函數。在這個模型中，除了要求每一條邏輯通道（logical channel）要其他邏輯通道正交（orthogonal）外，我沒有假設任何多工存取技術（multiple access scheme）。

由於這是一個 NP-hard 問題，我提出一個疊代的流程（iterative algorithm）和一個貪婪的策略（greedy algorithm）以提供一個接近最佳的解答。此外，我亦探討了兩個特例。這些特例被證明了可以用多項式時間流程（polynomial-time algorithm）得出最佳解答。電腦模擬結果也顯示了本論文提出的策略可以減少滿足用戶需求所需的資源。

Acknowledgement

I would like to express my gratitude to all those who gave me the possibility to complete this thesis. I am grateful to the Almighty God who is my source of strength throughout my study. "The LORD is my shepherd, I shall not be in want." (Psalm 23:1)

I take this opportunity to thank my supervisor, Prof. Tat Ming Lok. I am deeply indebted to your stimulating suggestions and encouragement during my research. You did not only help my research work, but you also broadened my horizon.

I would like to thank my family members who has supported me all the time. The encouragement of my parents motivated me to face the difficulties in my research. The prayers from my brother provided the power for me in the research.

I want to thank my brothers and sisters in my cell group in the Shatin Baptist Church. The prayers, the sharings, the motivations and the encouragement are the crucial backup for my research. It is my pleasure to have about twenty angels who remind me, "My help comes from the LORD, the Maker of heaven and earth." (Psalm 121:2) Although I did not pray for each of you, each of you has prayed for my research many times.

I also want to thank my previous and present colleagues in the Information and Systems Laboratory and Wireless Communications Laboratory II. Your comments are valuable in this thesis. Thank you for spending your precious time to answer my questions and providing suggestions.

This work is dedicated to the Almighty God and my family.

Contents

List of Figures

List of Tables

Preface

In the third and future generation wireless systems, operators and service providers have been introducing more and more variations of data applications. Examples include video conferencing, video streaming and various kinds of multimedia services. Quality of service (QoS) requirements have larger variations than previous generation wireless systems which only provide voice services. Therefore, more sophisticated resource allocation schemes are needed so that QoS requirements are satisfied in a more effective way. In this thesis, we consider the allocation of one important type of wireless resource, which is wireless channel.

Many current channel assignment schemes [19][20][39] have been proposed to optimize throughput or latency in many wireless data networks. However, client demands or QoS requirements are ignored in these schemes. As a result, unfair and inefficient assignment is resulted. Some clients may not be able to obtain their desired throughput while some may obtain far beyond their needs. Some channels are wasted in this case.

Some other schemes like [2] are proposed to improve the above situation by formulating user requirements as system constraints. However, in practice, there may not be enough number of channels or the channels do not have the desired high quality (like low signal-to-noise ratio). In the operations research terminology, there may be no feasible solutions.

One possible way to alleviate this problem is to perform admission control. Admitted data streams are guaranteed to meet the QoS requirements [36]. Then, we perform the throughput maximization or latency minimization for these admitted data streams. Another way to tackle this problem is to seek for a compromise solution. In this case, user requirements are no longer modelled as the system constraints of our problem. Instead, for each user, a function, called *unsatisfactory function*, is introduced which measures the deviation of the performance below the minimum requirement. By minimizing the sum of these functions, a compromise solution is obtained. In this compromise solution, on average, the performance of each client is close to the minimum requirement. This approach is more flexible so we consider it throughout this thesis.

In this thesis, we propose a *goal programming* [12] model for a general downlink channel assignment scheme. Goal programming is an operations research technique in seeking for compromise solutions. In our model, it does not only formulates the channel properties, but also introduces two functions, namely, the *unsatisfactory function* and the *bonus function*, to model the clients' QoS requirements and performance. For each user, these functions are chosen based on their application layer specification. In this model, there are no assumptions in the underlying multiple access schemes. It can be time division multiple access (TDMA), frequency division multiple access (FDMA), code division multiple access (CDMA) or the hybrid of them. The only requirement is that each logical channel must be orthogonal to one another. That means, the multiple access interference (MAI) is 0. Therefore, this model is general enough for most downlink transmission systems with a large variety of user applications.

In this thesis, it is shown that the problem is an NP-hard problem [11]. Hence, we propose two near-optimal polynomial-time algorithms, namely, *channel-swapping algorithm* and the *best-first-assign algorithm*. Simulation results show that our proposed algorithms assign the channels in a more effective way than throughput optimization schemes. Fewer channels are required to meet the same set of QoS requirements. This suggests our proposed algorithms are more economical than the throughput optimization approaches. In addition, in this thesis, we also compare these two proposed algorithms in terms of weighted sum of unsatisfactory function and time complexity. These are the two main concerns for the operators. This provides the hints on the choice of algorithm.

Our work does not end here. We also study two common subsets of problems where optimal solutions can be obtained by polynomial-time algorithms. In the first subset of problems, it is assumed that the *order of selection diversity* of the multiple access scheme is 1. On the other hand, in the second subset of problems, it is assumed that each client can be assigned at most 1 channel. Simulations results show that in the first subset of the problem, compared with the throughput optimization scheme, our proposed algorithm does not only have much smaller weighted sum of unsatisfactory function, but also the throughput of our proposed algorithm is close to the throughput optimization scheme. The proposed algorithm in the second subset of problem is used in obtaining a lower bound of weighted sum of unsatisfactory function in the performance evaluation of channel-swapping algorithm and best-first-assign algorithm.

Finally, we carry out the performance evaluation via simulations. As mentioned above, we compare our proposed algorithms and the throughput optimization schemes. In the simulations, the proposed algorithms outperform the throughput optimization scheme. We end this thesis with a conclusion and some future research directions.

This thesis is organized as follows. In Chapter 1, basic knowledge about multiple access schemes and goal programming are introduced. In chapter 2, previous works about channel assignment are reviewed. In Chapter 3, the general formulation for the channel assignment problem is proposed. Two algorithms are proposed accordingly. In Chapter 4, the two special cases are investigated and the optimal algorithms are proposed. In Chapter 5, performances of proposed algorithms in Chapter 3 and 4 are analyzed through simulations. In Chapter 6, I will conclude the thesis and discuss some future research directions.

Chapter 1

Introduction

The main theme of this thesis is the application of *goal programming* in channel assignment problem. Goal programming is a multi-objective optimization technique which is useful in seeking for compromise solution when no feasible solutions exist. Before we can formulate the problem as a goal programming model, we need to know how the spectrum is partitioned into different channels. This is the known as the *multiple access scheme.*

In this chapter, we will go through the two fundamental concepts of this thesis, namely, multiple access and goal programming. In multiple access, we will describe some common multiple access schemes like TDMA, FDMA, CDMA, etc. For goal programming, we will discuss how a problem can be formulated as a goal programming model. An example is given in the end of this chapter to illustrate the idea.

1.1 Multiple Access

In wireless communications, operators should allow multiple users to transmit and receive information simultaneously in a shared spectrum. This is the purpose of multiple access schemes. The base station (or access point for the case in wireless LAN) multiplexes all the data streams in a way that each user should be able to extract his or her desired data stream from the signal in the spectrum. In a multiple access scheme, the whole spectrum is divided into several *logical channels*. Each logical channel is dedicated to a one-way transmission (either uplink or downlink) between the base station and a client. In this thesis, for simplicity, the term *channel* refers to a logical channel of a multiple access scheme.

There are three major types of multiple access schemes which are adopted in many wireless networks. They are *time division multiple access* (TDMA), *frequency division multiple access* (FDMA) and *code division multiple access* (CDMA). Table 1.1 shows different multiple access techniques adopted in different wireless communication systems [29].

Table 1.1: Multiple Access Techniques Used in Different Wireless Communication Systems

Cellular System	Multiple Access Technique
Advanced Mobile Phone System (AMPS)	FDMA/FDD
Global System for Mobile (GSM)	TDMA/FDD
US Digital Cellular (USDC)	TDMA/FDD
Pacific Digital Cellular (PDC)	TDMA/FDD
CT2 (Cordless Telephone)	TDMA/FDD
Digital European Cordless Telephone (DECT)	FDMA/TDD
US Narrowband Spread Spectrum (IS-95)	CDMA/FDD
W-CDMA (3GPP)	CDMA/FDD
CDMA2000 (3GPP2)	CDMA/FDD

1.1.1 Time Division Multiple Access

In time division multiple access (TDMA) systems, the spectrum is divided into a set of nonoverlapping time slots. Each time slot is a logical channel. Hence, in each time slot, only one user can transmit or receive information.

In TDMA system, data transmission is not continuous but it is bursty. The advantage of this is the low power consumption of the mobile unit because during other users' time slots, the transmitter and receiver can be switched off. However, the cost is the synchronization overhead. Guard time is needed so that the transmitter and receiver are synchronized.

1.1.2 Frequency Division Multiple Access

In frequency division multiple access (FDMA) systems, the spectrum is divided into a number of frequency bands. Each frequency band corresponds to a logical channel. Therefore, different users can transmit or receive information in different frequency bands simultaneously.

Unlike TDMA, in which all users can use the whole spectrum during their data transmission, in FDMA, the bandwidth of each channel is much smaller. The symbol time of narrowband signal is large compared to the average delay spread. Thus, the intersymbol interference (ISI) is low and hence, the system does not require sophisticated equalization techniques. Nonetheless, FDMA systems require tight radio frequency (RF) filtering to minimize adjacent channel interference.

1.1.3 Code Division Multiple Access

In code division multiple access (CDMA) systems, each logical channel is a *signature sequence* (also known as *spreading sequence*). The correlation between the signature sequences is low and by using this property, receivers can differentiate signals of different logical channels by either the matched filter [27] (for zero correlation, i.e. orthogonal sequences) or multiuser detectors [37] (for non-zero correlation, i.e. non-orthogonal sequences).

There are many ways to implement CDMA. Two important examples of CDMA systems are *direct-sequence CDMA* (DS-CDMA) systems [26] and *multicarrier CDMA* (MC-CDMA) [21] systems.

In DS-CDMA systems, transmitted signals of different users are multiplied by different *spreading signal.* The spreading signal is of the form:

$$\sum_{l=0}^{N} a_l \psi(t - lT_c) \tag{1.1}$$

where $\{a_l\}$ is the signature sequence and $\psi(t)$ is the *chip waveform*, which is time-limited to $[0, T_c)$ where T_c is the chip interval. Since the correlation between distinct sequences is low, to extract the desired signal, we can adopt a correlator receiver [42] to multiply and integrate the received signal with the spreading signal.

On the other hand, in MC-CDMA systems, instead of multiplying the user signal by the spreading signal in the time domain, the multiplication is carried out in frequency domain. As its name implies, in MC-CDMA systems, every user makes use of all the carriers in the system. Each carrier is orthogonal to one another. The user signal is multiplied by every carrier. For each carrier, the modulated signal is multiplied by an element of the signature sequence. The transmitted signal is the aggregation of the modulated signals of all the carriers.

1.1.4 Hybrid Multiple Access Scheme

Apart from the above multiple access schemes, there are some schemes called *hybrid multiple access schemes.* They are combinations of multiple access schemes. After dividing the spectrum into logical channels by the first multiple access scheme, each logical channel is further divided into a new set of logical channels by the second multiple access scheme.

One example is the *time division CDMA* (TCDMA) [29] system. In TCDMA system, different signature sequences are assigned to different set of users. Within the same set of users, only one user can transmit or receive the signal in a time slot with that signature sequence. In this case, a logical channel is the signature sequence in a time slot.

1.2 Goal Programming

In many common optimization models, a single objective function is optimized. The optimization may be performed subject to a certain set of constraints. However, in reality, we may encounter problems involving more than one objective function which may conflict with one another. Furthermore, feasible solutions may not exist. In this situation, we may interest in seeking some *compromise solutions* so that the final decision is close to the minimum requirements of everybody involved. Therefore, we need some formulations which facilitate us to seek for compromise solutions in these scenarios.

To alleviate the above problems, Charnes and Cooper proposed an approach called *goal programming* in [5]. The principal idea is to combine all the objective functions and the *soft* constraints into a single objective function. Soft constraints refer to those constraints that we should try our best to fulfill but we are allowed to provide solutions which do not satisfy these constraints. For example, in some network resource allocation problems, every user's application has its own quality of service (QoS) requirements according to the application layer specification. If there is available amount of resource, those QoS requirements should be satisfied. However, in reality, there exist some situations that the system does not have enough resource for the QoS requirements. Therefore, to have a more realistic formulation, those QoS requirements should be modelled as soft constraints. Actually, this is one important point in the channel assignment formulation proposed in this thesis.

After that, we transform all the soft constraints in the following way. For the i-th soft constraint, we define a function $f_i(\vec{x})$, where $\vec{x}$ is the vector of decision variables. This function f_i is a measurement of deviation from the requirement of the constraint. For example, if constraint i is $g_i(\vec{x}) \leq a_i$, where a_i is a constant, one possible choice of f_i can be $f_i(\vec{x}) = \max\{0, g_i(\vec{x}) - a_i\}$. That means, if this constraint is satisfied, $f_i(\vec{x})$ is 0. Otherwise, it is $g_i(\vec{x}) - a_i$, that is the deviation of $g_i(\vec{x})$ from its required upper limit.

Now, the soft constraints become functions of decision variables. Since the functions $\{f_i\}$ are deviations from the problem requirement, we would like to minimize them. Therefore, for each f_i, we create a new objective function:

$$\text{Minimize} \quad f_i(\vec{x}). \tag{1.2}$$

Together with the above set of new objective functions, we now have a *multi-objective* optimization problem. In general, these constraints may conflict with one another. Thus, we need to combine them into a single objective function in the following manner first.

To begin with, we classify the objective functions into different groups according to their priorities in decision making. A higher priority objective function dominates the lower priority objective function in decision making. The objective functions in the same group do not dominate one another in decision making. Usually, the objective functions for the soft constraints are in the highest priority [12].

Next, we define a term called *lexicographic minimum* as below [12]:

Definition 1.1. *For two vectors $\vec{a}^{(1)} = (a_1^{(1)}, a_2^{(1)}, \ldots, a_n^{(1)})^T$ and $\vec{a}^{(2)} = (a_1^{(2)}, a_2^{(2)}, \ldots, a_n^{(2)})^T$, $\vec{a}^{(1)}$ is preferred to $\vec{a}^{(2)}$ if there exists an integer k such that $a_k^{(1)} < a_k^{(2)}$ and all higher order terms (i.e. $a_1, a_2, \ldots, a_{k-1}$) are equal. If no other vectors is preferred to $\vec{a}$, then $\vec{a}$ is the lexicographic minimum.*

With the ranking of the objective functions and Definition 1.1, we can combine the objective functions into a single objective function as follows. Firstly, the objective functions in the same priority group is combined linearly into one objective function. The coefficients in this linear combination correspond to the relative importance of the objective functions in decision making. Now, for each priority group, there is one combined objective function. We put all these objective functions into a vector. The first component is for the most important objective function, the second component is for the second most important objective function and so on. Our final objective function becomes:

$$\text{lexmin} \quad (g_1(\vec{x}), g_2(\vec{x}), \ldots)^T \tag{1.3}$$

where g_i is the objective function for the i-th priority group and $\vec{x}$ is the vector of the decision variables.

According to the properties of the objective functions, there are different algorithms to solve the goal programming problem. For example, if all the components are linear functions, the problem can be solved by multiphase Simplex-method [12], which is an extension to the two-phase method [10] used in linear programming problems. For nonlinear goal programming problems, approaches have been summarized in [32].

We end this chapter with the following example quoted from [35] to illustrate how to formulate a goal programming model. Since this thesis is not dedicated to the topic goal programming, interested parties may refer to [12] for further details.

Example 1.1. *Fairville is a small city with a population of about 20,000 residents. The city council is in the process of developing an equitable tax rate table. The annual taxation base for real estate property is \$550 million. The annual taxation bases for food and drugs and for general sales are \$55 million and \$35 million, respectively. Annual local gasoline consumption is estimated at 7.5 million gallons. The city council wants to maximize the tax revenue by developing the tax rates based on three main goals*

- *Food and drug taxes cannot exceed 10% of all taxes collected.*
- *General sales taxes cannot exceed 20% of all taxes collected.*
- *Gasoline tax cannot exceed 2 cents per gallon.*

Let the variables $x_p, x_f,$ *and* x_s *represent the tax rates (expressed as proportions) for property, food and drug, and general sales and define the variable* x_g *as the gasoline* *tax in cents per gallon. The problem can be formulated as*

$$Maximize \quad 500x_p + 35x_f + 55x_s + 0.075x_g \tag{1.4}$$

subject to

$$35x_f \leq 0.1(550x_p + 35_xf + 55x_s + 0.075x_g) \tag{1.5}$$

$$55x_s \leq 0.2(550x_p + 35_xf + 55x_s + 0.075x_g) \tag{1.6}$$

$$x_g \leq 2 \tag{1.7}$$

$$x_p, x_f, x_s, x_g \geq 0 \tag{1.8}$$

Each of the first three inequalities represents a goal that the city council aspires to satisfy. However, these goals may be in conflict and the best we can do is try to reach a compromise solution.

Firstly, we convert the first three inequalities as follows:

$$55x_p - 31.5x_f + 5.5x_s + 0.0075x_g + s_1^+ - s_1^- = 0 \tag{1.9}$$

$$110x_p + 7x_f - 44x_s + 0.015x_g + s_2^+ - s_2^- = 0 \tag{1.10}$$

$$x_g + s_3^+ - s_3^- = 0 \tag{1.11}$$

$$s_i^+, s_i^- \geq 0 \quad i = 1, 2, 3 \tag{1.12}$$

These constraints replace the old set of system constraints. In addition, we have the following three new objective functions now:

$$Minimize \quad s_1^+ \tag{1.13}$$

$$Minimize \quad s_2^+ \tag{1.14}$$

$$Minimize \quad s_3^- \tag{1.15}$$

These three objective functions dominate the objective function (1.4) in decision making. Furthermore, these three objective functions do not dominate one another in decision making and they are equally important. Therefore, these objective functions are combined into the following single objective function:

$$lexmin \quad \left(s_1^+ + s_2^+ + s_3^-, 500x_p + 35x_f + 55x_s + 0.075x_g\right)^T \tag{1.16}$$

subject to the constraints (1.9) to (1.12).

Chapter 2

Previous Works in Channel Assignment

In this chapter, we will review some previous works in channel assignment. We will discuss the disadvantage of these channel assignment schemes and the rationale behind as a background study and motivation of proposing new channel assignment models and schemes in this thesis.

2.1 Voice Service Network

In the second generation (2G) cellular network, only voice service is provided. Hence, if the wireless channel has a signal-to-noise ratio (SNR) above certain threshold, it can be assigned to a user. Thus, for every user, all channels above the SNR threshold are identical. The base station only needs to assign any of these channels which is available. This is the channel assignment scheme in 2G systems.

The grade of service (GOS) is the blocking probability of the network. There are two types of trunked systems which have two different formulas for the GOS. The first type offers no queueing for the call requests. For each user who requests service, it is assumed there is no setup time. If a channel is available, the suer can access it immediately. Otherwise, that user is blocked without access and is free to try again later. The inter-arrival time of the users is assumed to be Poisson distributed and the service time of each user is assumed to be exponentially distributed. The problem is modelled by an M/M/C queueing system [31]. In this case, the GOS is given by the Erlang B formula [3]

$$Pr\left\{\text{blocking}\right\} = \frac{\frac{A^C}{C!}}{\sum_{k=0}^{C} \frac{A^k}{k!}} \tag{2.1}$$

where C is the number of channels in the cell and A is the offered load, which is the product of mean arrival rate and mean service time.

In the second type trunked system, a queue is provided to hold the blocked calls. Call requests are delayed until a channel is available. In this case, the GOS is defined as the probability that a call is blocked after waiting a specific length of time in the queue. Before determining the GOS, the probability that a call not having an immediate access to a channel is determined by the Erlang C formula [29]

$$Pr\left\{\text{delay} > 0\right\} = \frac{A^C}{A^C + C!\left(1 - \frac{A}{C}\right)\sum_{k=0}^{C-1} \frac{A^k}{k!}} \tag{2.2}$$

The probability that a delayed call is forced to wait more than t seconds is given by the probability that the call is delayed, multiplied by the conditional probability that the delay is greater than t seconds. Hence, the GOS is given by

$$Pr\left\{\text{delay} > t\right\} = Pr\left\{\text{delay} > 0\right\} Pr\left\{\text{delay} > t | \text{delay} > 0\right\} \tag{2.3}$$

$$= Pr\left\{\text{delay} > 0\right\} e^{-\frac{(C-A)t}{H}} \tag{2.4}$$

where H is the mean service time.

2.2 Data Network

In the third generation and future generation wireless networks, there are not only the voice services but also more and more data applications. For data applications, common choices of performance measures are throughput and latency [28]. The quality of service (QoS) of the network is no longer the blocking probability only.

According to Shannon's Channel Coding Theorem [33], for a channel with bandwidth B and signal-to-noise ratio (SNR) γ, there exists a channel coding scheme such that the coding rate is any C, $C \leq R$, where R is given by

$$R = B \log_2(1 + \gamma). \tag{2.5}$$

This theorem implies that for a given amount of bandwidth, the maximum achievable throughput of a channel depends on its SNR. In general, each channel has different SNR. It is because the fading experience of each channel is different. In addition, for the same channel, the fading experience of different users is different. Fading affects the received signal power and thus the SNR at the receiver side.

For some multiple access schemes, for the same user, the fading experience of different channels is the same but this is not the case for other multiple access schemes. For instance, in a narrowband direct sequence CDMA (DS-CDMA) system with Hadamard signature sequences [38], for the same user, the fading experience of different channels is the same [20]. On the other hand, if we use random orthogonal signature sequences, for the same user, the fading experience of different channels is different.

We will define a term *order of selection diversity* in Chapter 3 for this phenomenon. This is a key property in designing special case algorithms in Chapter 4

By assigning different channels, the data applications may have different performance because the throughputs of different channels are different. Unlike voice service networks, we should not assign an available channel arbitrarily because the performance measure of the system is no longer the blocking probability. In this case, channel estimations have to be performed and then channels are assigned based on these estimated values. The channels are assigned to optimize certain performance measures. Some channel assignment schemes in data networks are reviewed below.

2.2.1 Throughput Optimization

Since throughput and latency are typical choices of performance measures, some channel assignment schemes [19][20][39] are proposed to optimize these two measurements in different networks or multiple-access schemes. Since latency is the reciprocal of the throughput, for channel assignment schemes which only assign one channel to each user like [20], the same scheme also minimizes the total latency of the users.

We can have a general formulation for the throughput optimization scheme. Let N and K be the number of channels and users respectively. Let $x_{i,j}$ be the binary decision variable such that it is 1 if channel i is assigned to client j. Otherwise, $x_{i,j}$ is 0. Let $R_{i,j}$ be the throughput of channel i for client j, which is obtained by equation (2.5). The formulation of the problem is as follows.

$$\text{Maximize} \quad \sum_{i=1}^{N}\sum_{j=1}^{K} R_{i,j}x_{i,j} \tag{2.6}$$

subject to

$$\sum_{i=1}^{N} x_{i,j} \leq 1, \quad \forall j \tag{2.7}$$

$$\sum_{j=1}^{K} x_{i,j} \leq 1, \quad \forall i \tag{2.8}$$

$$x_{i,j} \in \{0,1\} \tag{2.9}$$

where the objective function is the total throughput of the users.

It can be seen that it is an *assignment problem* [10] and it can be solved by the *Hungarian method* [16]. Alternatively, it can also be solved by common mathematical software.

The major drawback of these schemes is that the application layer specifications, such as minimum required throughput, are not considered. An unfair channel assignment may be resulted. The reason is that to optimize the total throughput, the base station tends to assign more channels to users with higher average SNR.

Consequently, some users obtain more channels than they need because they have relatively high SNR on average but their data application does not require high throughput. On the other hand, some users do not have enough channels because they have relatively low SNR on average but their data application requires very high throughput. In this case, some channels are wasted. To assign the channels more efficiently, more sophisticated schemes that consider QoS requirements are needed.

2.2.2 Channel Assignment Schemes with QoS Consideration

To remedy the problem of throughput optimization, that is to satisfy the user application requirements, some channel assignment schemes [2][7] optimize the throughput or latency subject to certain QoS requirements. In this case, QoS requirements are modelled as system constraints. For example, we may add the following constraints to the model in section 2.2.1:

$$\sum_{i=1}^{N} R_{i,j} x_{i,j} \geq r_j, \quad \forall j \tag{2.10}$$

where the left hand side of the inequality is the total throughput of user j and r_j is the minimum required throughput of user j.

Hence, all feasible solutions are guaranteed to fulfill the QoS requirements so the problem of pure throughput optimization is solved. These channel assignment schemes choose the feasible solution which has the highest throughput. Nonetheless, the system may not have enough number of channels to satisfy all the users' QoS requirements. As a result, it is possible that there may not be any feasible solutions.

There are two approaches to solve this problem. The first method is to have a stringent admission control policy [4][19] to admit part of downlink flows. The admitted set of flows are guaranteed to fulfill the QoS requirements of the user applications. Then, we assign the channels for these admitted flows so that the throughput is optimum by using the above algorithms.

The second method is to seek for a compromise solution instead of applying a stringent admission control policy. In this method, although it is impossible to fulfill the user requirements, on average, the performance is very close to the minimal QoS requirements. In this case, QoS requirements are met as much as possible. In this thesis, the proposed channel assignment schemes adopt the second approach as it is a more flexible approach.

As mentioned in Chapter 1, one purpose of using goal programming is to seek for a compromise solution when there are no feasible solutions. Therefore, in the second approach, goal programming is a candidate for tools of the second approach.

Chapter 3

General Channel Assignment Scheme

In section 2.2.2, it mentioned two approaches of channel assignment to deal with the case that there is no feasible solutions. One approach is to seek for a compromise solution so that on average, the performance is close to the minimal QoS requirements.

In section 1.2, it introduced an optimization technique known as goal programming to seek for compromise solutions when no feasible solutions exist. In this chapter, we propose a goal programming model for a general downlink channel assignment scheme. This model not only formulates the channel properties, but also introduces two functions, the *unsatisfactory function* and the *bonus function*, to model the clients' requirements and performance. For each user, these functions are chosen based on the user application specification. In this model, there is no assumptions for the underlying multiple access schemes. The only requirement is that each logical channel must be orthogonal to one another. That means, the multiple access interference (MAI) is 0. Therefore, this model is general enough for most downlink transmission systems with a large variety of user applications.

It is shown that the problem is an NP-hard problem, so we propose two near-optimal polynomial-time algorithms, the *channel-swapping algorithm* and the *best-first-assign algorithm*, based on this formulation. The content of this chapter is also published in [25].

3.1 Baseline Model

We consider a system with N channels and K clients. Here, the meaning of a client is not constrained to be a single user with his/her mobile terminal. For a single user, if he/she has more than one user applications which require parallel and independent transmissions, we can also model each user application as an independent client. On the other hand, in chapter 4, in some special cases, we would like to combine a group of clients into one virtual client to reduce the computational time of the channel assignment scheme. Another example of combining a group of clients is in example 3.3.

Let $x_{i,j}$ be a binary decision variable such that

$$x_{i,j} = \begin{cases} 1, \text{if channel } i \text{ is assigned to client } j \\ 0, \text{otherwise} \end{cases} \tag{3.1}$$

For client j, due to the system constraint of the mobile unit of him/her, at most n_j channels can be assigned to him/her. If there is no such constraint for that user, n_j can be set to N, which is the number of channels of the whole system. However, in many current communication system, n_j is equal to 1. On the other hand, each channel can be assigned to at most one client. Thus, we obtain the following two system constraints:

$$\sum_{i=1}^{N} x_{i,j} \leq n_j \quad \forall j \tag{3.2}$$

$$\sum_{j=1}^{K} x_{i,j} \leq 1 \quad \forall i \tag{3.3}$$

The first inequality means that user j can have not more than n_j channels. The second inequality means that each channel cannot be assigned to more than 1 user.

We assume that the channels are orthogonal, i.e. multiple access interference (MAI) is zero. For each channel i to each client j, we define a value called *quality index*, $R_{i,j}$. This value describes the quality of the i-th channel enjoyed by the j-th client. The choice of value depends on the client's application. One example of this value is the throughput obtained by client j when channel i is assigned to him/her. This can be the quality index of a channel for a video conferencing application. Another example of this value is the SNR of that channel. This can be the quality index of a channel for a voice communication. The higher the quality index, the better the channel.

For each of client j, there are two functions associated with him/her, the *unsatisfactory function*, $d_j^+(\sum_{i=1}^N R_{i,j}x_{i,j})$, and the *bonus function*, $d_j^-(\sum_{i=1}^N R_{i,j}x_{i,j})$. The unsatisfactory function is a monotonic decreasing function of $\sum_{i=1}^N R_{i,j}x_{i,j}$ while the bonus function is a monotonic increasing function of $\sum_{i=1}^N R_{i,j}x_{i,j}$. These two functions are the specifications of the user applications. Briefly speaking, the unsatisfactory function is a measurement of the performance below the user's minimum requirement for a given channel assignment. On the other hand, the bonus function is a measurement of the performance beyond the user's minimum requirement. Similar to the quality index, the choice of explicit form unsatisfactory function and bonus function depends on the application layer requirement of that user. Below are three simple examples to illustrate the choice and the meaning of both functions.

Example 3.1. *We consider a multi-carrier CDMA (MC-CDMA) system [21] with orthogonal signature sequences [38]. Since the signature sequences are orthogonal to one another, the MAI is zero.*

The i-th signature sequence is denoted by $\vec{s_i}$. Each sequence is normalized to unit norm. Let g_j be the large path loss of client j. The background noise is assumed to be the additive white Gaussian noise (AWGN) with zero mean and variance σ^2. The

SNR of channel i of client j is given by [20]

$$\gamma_{i,j} = \frac{g_j^2}{\sigma^2 \vec{s_i}^H \mathbf{A}_j^{-1} \mathbf{A}_j^{-1H} \vec{s_i}} \tag{3.4}$$

where $\alpha_{i,j}$ is a factor which accounts for the overall effects of phase shift and fading for the ith carrier of the jth client's receive signal and $\mathbf{A}_j$ is a diagonal matrix whose ith element is $\alpha_{i,j}$. The SNR of a channel is obtained by channel estimation schemes.

Suppose client j would like to download some data. In this case, one obvious choice of quality index of a channel is the throughput of that channel. The throughput of channel i is chosen to be the value of $R_{i,j}$. Let B be the channel bandwidth. By Shannon's channel coding theorem [33], there exists a channel coding scheme such that the throughput of that channel is $R_{i,j} = B\log_2(1+\gamma_{i,j})$. We choose this value as the quality index.

Assume client j demands a minimum throughput of D_j. The unsatisfactory function and bonus function can be chosen as:

$$d_j^+(\sum_{i=1}^{N} R_{i,j}x_{i,j}) = \max\left\{D_j - \sum_{i=1}^{N} R_{i,j}x_{i,j}, 0\right\} \tag{3.5}$$

$$d_j^-(\sum_{i=1}^{N} R_{i,j}x_{i,j}) = \max\left\{\sum_{i=1}^{N} R_{i,j}x_{i,j} - D_j, 0\right\}. \tag{3.6}$$

In this case, the unsatisfactory function is the amount of throughput below the minimum required throughput. The bonus function is the amount of throughput above the minimum required throughput.

On the other hand, if client j only requires a channel irrespective of the channel throughput, he/she may choose $R_{i,j}$ to be 1 and the unsatisfactory function and bonus function to be:

$$d_j^+(\sum_{i=1}^{N} R_{i,j}x_{i,j}) = \max\left\{1 - \sum_{i=1}^{N} x_{i,j}, 0\right\} \tag{3.7}$$

$$d_j^-(\sum_{i=1}^{N} R_{i,j}x_{i,j}) = 0 \tag{3.8}$$

An example of it is the traditional voice service. The user only requires a channel with high enough SNR.

Example 3.2. *This example shows how our model can generalize some previous models channel assignment problems. In this example, we would like to use our model for throughput optimization.*

To perform the throughput optimization, the unsatisfactory function and bonus function can be chosen as below:

$$d_j^+(\sum_{i=1}^{N} R_{i,j}x_{i,j}) = 0 \tag{3.9}$$

$$d_j^-(\sum_{i=1}^{N} R_{i,j}x_{i,j}) = \sum_{i=1}^{N} R_{i,j}x_{i,j} \tag{3.10}$$

where $R_{i,j}$ is chosen as the throughput of channel i for user j. In throughput optimization problem, we do not have any throughput requirement so we choose the unsatisfactory function to be 0. Then, we choose the bonus function as the total throughput of that user.

For latency minimization, you may choose $R_{i,j}$ to be the negative of latency of channel i for user j and the same unsatisfactory function and bonus function. Alternatively, you may choose $R_{i,j}$ to be the latency of channel i for user j. Then, the unsatisfactory function and bonus function can be chosen as:

$$d_j^+(\sum_{i=1}^{N} R_{i,j}x_{i,j}) = \sum_{i=1}^{N} R_{i,j}x_{i,j} \tag{3.11}$$

$$d_j^-(\sum_{i=1}^{N} R_{i,j}x_{i,j}) = 0 \tag{3.12}$$

Example 3.3. *In this example, we consider the case of broadcasting a signal to a group of users. Suppose users $1, 2, \ldots, m$ would like to watch the same stream of video. We would like these users' mobile terminal to listen to the same set of channels so that more channels can be assigned to other users. In this case, we can group all the users into one virtual user. Since the users are using the same* *application, the explicit form of the unsatisfactory function and bonus function is the same for all of them except the quality indices due to different fading experience of the users. Then, the choice of the quality index of each channel i of this virtual user can be the minimum of the quality index of the corresponding group of users. The choice of unsatisfactory function and bonus function of this virtual user is the same as the ones of each member of that group of user.*

As shown in the above examples, for different types of client applications, the choices of quality index, unsatisfactory function and bonus function are different. In addition, from the above example, it can be seen that for client j, the value of $R_{i,j}$ may or may not depend on i. In some special cases, such as using Hadamard signature sequence in narrowband DS-CDMA systems, the SNR of each channel is the same so $R_{i,j}$ is the same for all i. In the second case, the value of $R_{i,j}$ is clearly independent of i if the transmitted power of each channel is very high. In [20], the term *order of selection diversity* is defined. We can further generalize the definition as below:

Definition 3.1. *For client j, the* order of selection diversity *of client j is the number of distinct values in the set $\{R_{i,j} : 1 \leq i \leq N\}$.*

If we choose $R_{i,j}$ to be the throughput of channel i of client j, we can obtain the same meaning of order of selection diversity as [20]. However, in definition 3.1, the order of selection diversity does not only depend on the fading characteristics, but also depends on the client application. This property is crucial when we try to seek for special case algorithms in chapter 4.

Now, for client j, there are two objective functions:

$$\text{Minimize} \quad d_j^+(\sum_{i=1}^{N} R_{i,j}x_{i,j}) \tag{3.13}$$

$$\text{Maximize} \quad d_j^-(\sum_{i=1}^{N} R_{i,j}x_{i,j}) \tag{3.14}$$

Now, we have an optimization problem with $2K$ objective functions. In general, this set of $2K$ objective functions may conflict with each other. This baseline model is then transformed to a goal programming model to resolve this conflict.

3.2 Goal Ranking

To begin with, we rank the objective functions of the clients. The objective functions are divided into several priority classes. In chapter 1, it says that an objective function in a higher priority class dominates another one in a lower priority class in decision making.

Since the unsatisfactory function is the deficiency of performance below the minimum requirement of the user, minimizing the unsatisfactory function is much more important than maximizing the bonus function. Therefore, the objective functions in (3.13) dominate the ones in (3.14). Thus, the objective functions in (3.13) is a higher priority class while the remaining ones are in a lower priority class. In this case, we divide the objective functions into two priority classes.

However, among the objective functions of each priority class, none of them dominates another objective function. Therefore, we do not further divide the priority classes. Hence, in our problem, we only have the two priority classes mentioned above.

3.3 Model Transformation

In section 1.2, we have defined the term *lexicographic minimum* in Definition 1.1. By using the model transformation technique in section 1.2 and the division of objective functions into priority classes in section 3.2, we can transform the baseline model into a goal programming model.

Now, the baseline model introduced in the section 3.1 can be converted into a single-objective model as follows. Firstly, we aggregate the objective functions in a priority class into a single objective function by a linear combination. Then, we place these two new objective functions (one for unsatisfactory functions and another one for bonus function) into a vector. The lexicographic minimum of this vector is the final objective function. Since we would like to maximize the bonus functions, a minus sign should be added to the weighted sum of them as minimizing the negative of the weighted sum of them is equivalent to maximizing the weighted sum of them. So, we obtain the goal programming model for the channel assignment problem as below:

$$\text{lexmin} \quad \vec{u} = (\sum_{j=1}^{K} \beta_j d_j^+(\sum_{i=1}^{N} R_{i,j}x_{i,j}), -\sum_{j=1}^{K} \beta_j d_j^-(\sum_{i=1}^{N} R_{i,j}x_{i,j}))^T \tag{3.15}$$

subject to

$$\sum_{i=1}^{N} x_{i,j} \leq n_j \quad \forall j \tag{3.16}$$

$$\sum_{j=1}^{K} x_{i,j} \leq 1 \quad \forall i \tag{3.17}$$

$$x_{i,j} \in \{0,1\} \tag{3.18}$$

where β_j are predefined constants.

The new objective function in equation (3.15) is called the *achievement function* in goal programming terminology. Inside the achievement function, the values β_j denotes the relative importance of user j's objective function. For instance, if client j pays for a more expensive service plan, his/her value of β_j will be higher.

In addition, these constants can also be used to ensure some fairness criteria. For example, if a user, on average, has relatively high value of the unsatisfactory function compared to other users, we may assign a higher value of β_j to him/her.

3.4 Proposed Algorithms

Proposition 3.1. *The optimization problem formulated in equations (3.15) to (3.18) is an NP-hard problem [11].*

The proof of this proposition is provided in Appendix A. Briefly speaking, one special case of our problem is the *number partitioning problem* which has been shown to be NP-complete [9].

Since this problem is shown to be an NP-hard problem, for practicality, we seek for near-optimal polynomial-time algorithms in this chapter.

3.4.1 Channel Swapping Algorithm

As shown above, the optimization problem is NP-hard. Therefore, we propose a suboptimal scheme called *channel swapping algorithm.* This algorithm is divided into two parts. It is outlined as follows. Firstly, we obtain an initial feasible solution by an arbitrary assignment. Then, we try to improve this solution by swapping assignments between users in each iteration.

In the second part of the algorithm, we do not only consider pairwise swapping, but we also consider a sequence of swapping so that for each iteration, the improvement is higher. The algorithm of the second part is similar to the shortest path algorithm. Each channel is represented by a node and the change by swapping the assignment of two clients in the objective function is the distance between each pair of the node. Each node would keep its own information about the channel assignment. The distance between each pair of node is updated according to the current information available in each node. The starting channel is the initial node. And we compare the total change of objective δD to this initial node. The information would be updated according to δD. For each channel, the following steps are executed in each iteration.

1. Set δD of each node to be infinite except the initial node.
2. Calculate the change in the objective if this channel is assigned to a different user.
3. If the new δD of a node is smaller, update the node's information including its assignment and its δD.

4. If the node is updated, go to step 2.

5. Stop when there are no updated node remain or a negative δD is detected in the initial node.

6. If a negative δD is found, change the assignment according to it.

At each iteration, we consider all the N nodes as the initial nodes. Therefore, we have N shortest path problems. For each of the shortest path problem, it can be solved by the *Dijkstra's Algorithm* [40]. Since the time complexity of the Dijkstra's Algorithm is $O(N^2)$ [40], the time complexity for the whole iteration is $O(N^3)$.

This iterative algorithm converges. It is because after each iteration, the weighted sum of unsatisfactory functions decreases and the channel assignment is still a feasible solution. That means, the weighted sum of unsatisfactory functions decrease and is bounded below by the optimal solution for every iteration. Therefore, this algorithm converges.

Figure 3.1 shows an example which illustrates the convergence of this algorithm. In this example, we consider a MC-CDMA system with 32 channels and 16 clients. The sequences are random orthogonal sequences. They are generated by a random matrix followed by the QR decomposition [15]. The row vectors are the orthogonal sequences. Each client can support at most 2 channels (i.e. $n_j = 2$). The unsatisfactory function and bonus function considered are equation (3.5) and (3.6) respectively. Then, we plot the sum of unsatisfactory function (in this example, we simply set β_j to be 1 for all j) against the iterations in figure 3.1 for the first 10 iterations.

It can be seen that the algorithm converges within 3 to 4 iterations. From the simulations results of other scenarios, it is found that typically, the algorithm converges within 4 to 5 iterations. This shows that the algorithm has a high convergence speed.

For the first part of the algorithm, instead of arbitrary assignment, we may also choose other methods to obtain an initial feasible solution. One possible candidate can be the solution from the throughput optimization scheme. However, from the

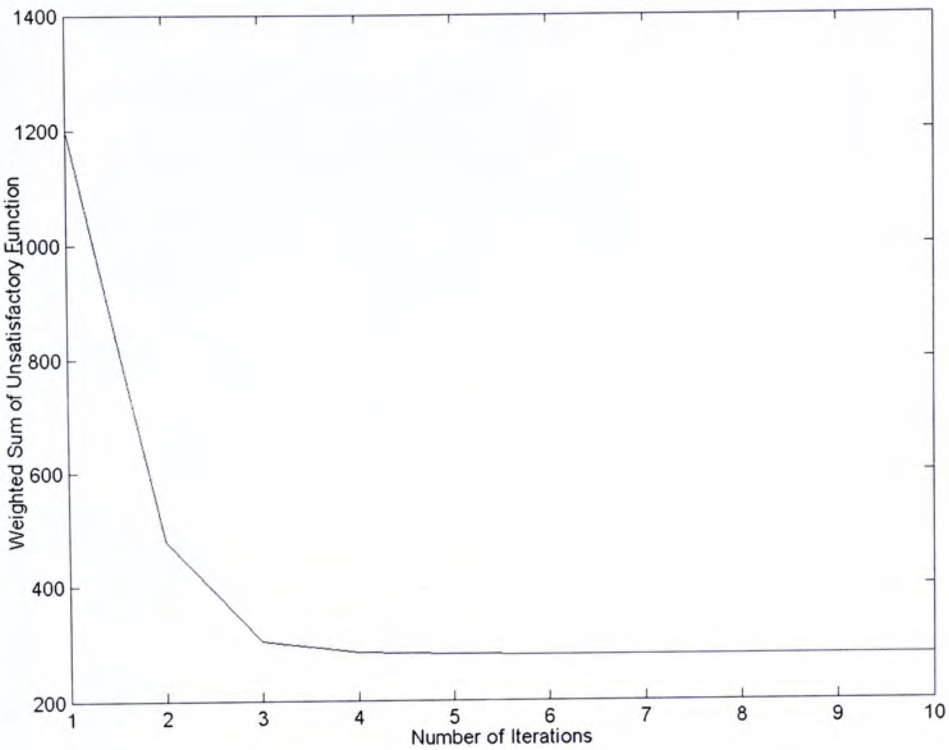

Figure 3.1: Relationship between weighted sum of unsatisfactory function and number of iterations

simulation, we found that the final performance of this candidate and arbitrary assignment, i.e. the weighted sum of unsatisfactory function, is approximately the same. Therefore, we suggest to use arbitrary channel assignment to obtain the initial feasible solution because of its smaller time complexity.

3.4.2 Best-First-Assign Algorithm

The iterative algorithm has the time complexity of $O(N^3)$ for each iteration. The computational complexity is high for the whole implementation although the convergence speed is very fast as shown in section 3.4.1. In this section, we provide another algorithm which requires smaller time complexity. A detailed comparison of the performance of these two algorithms will be given in Chapter 5.

In this section, we propose the scheme called *best-first-assign algorithm*, which is a greedy approach [6], as outlined below:

Step 1: For each client j, sort the value $R_{i,j}$ in descending order.

Step 2: For each client j, evaluate the reduction of d_j^+ if a unassigned channel with highest value of $R_{i,j}$ is assigned to him/her.

Step 3: Among these clients, choose the client has the greatest reduction calculated in step 2 and perform the corresponding channel assignment.

Step 4: If there are still some channels can be assigned to the clients, go to step 2.

In step 1, the sorting can be processed by parallel processing as each client's sorting list is independent from the others. The time complexity for the sorting is $O(N \log N)$ [6]. The time complexity for step 2 and 3 for each iteration is $O(K)$. There are at most N iterations as in each iteration, one channel is assigned. Therefore, the time complexity of this algorithm is $O(NK)$, which is smaller than the one of channel swapping algorithm. In chapter 5, it can be seen that the cost of having smaller time complexity is the increment of the weighted sum of unsatisfactory function. However, also in chapter 5, it shows that this increment is small. Hence, the best-first-assign algorithm is also a good choice for our general channel assignment problem.

Chapter 4

Special Case Algorithms

In Chapter 3, we have formulated a goal programming model for a general channel assignment problem. In this chapter, we will go through two special cases of this general channel assignment problem. The first case is that the order of selection diversity (see Definition 3.1) is 1. The second case is that every user can support at most one channel (i.e. $n_j = 1$ for all j).

In Chapter 3, we have mentioned that the general channel assignment problem is NP-hard. Nevertheless, for these two special cases, optimum solutions can be obtained by means of polynomial-time algorithms. This is the purpose of this chapter. The details of these two cases are provided below.

4.1 Single Order of Selection Diversity

In this section, we consider the case that the order of selection diversity of all users is 1. According to Definition 3.1, for a user j, the quality indices $R_{i,j}$ is the same for all channels. That means, for each user, all channels are the same. On the other hand, for each channel, different users experience different performances.

Such behavior is typical in the downlink of a cellular system. One example is a narrowband DS-CDMA system with Hadamard signature sequences [38]. In this case, the SNR of each signature sequence is the same. Another example is a special case of Example 3.1. If the network only provides voice services, $R_{i,j} = 1$ for all i and j. So the order of selection diversity of the system depends on the multiple access scheme and the user application. Therefore, for each user, he/she only concerns about how many channels are assigned to him/her. The system model in Chapter 3 can be simplified as below.

The content of this section is also published in [24].

4.1.1 System Model

Since for each client, the channels are identical, the problem is reduced to decide how many channels should be assigned to each client. That means, we need to change the decision variables. Let x_j be the number of channels assigned to client j. Thus, the unsatisfactory function and bonus function of a user j can be modelled as functions of x_j.

Now, the system model can be modified as below:

$$\text{lexmin} \quad \vec{u} = \left(\sum_{j=1}^{K} \beta_j d_j^+(x_j), -\sum_{j=1}^{K} \beta_j d_j^-(x_j) \right)^T \tag{4.1}$$

subject to

$$\sum_{j=1}^{K} x_j \leq N \tag{4.2}$$

$$0 \leq x_j \leq n_j \quad \forall j \tag{4.3}$$

$$x_j \in \mathbb{N} \quad \forall j \tag{4.4}$$

Constraint (4.2) means there are totally N channels which are available. Constraint (4.3) is the system constraint of the mobile terminal of each client which is the maximum number of channels can be assigned to client j. It also ensures that the decision variables are non-negative.

4.1.2 Proposed Algorithm

To solve the goal programming problem above, we propose the following algorithm, called *inductive assignment algorithm* which is a *dynamic programming* [6] approach.

Let $\vec{f}(n,k)$ be the optimal value of $\left(\sum_{j=1}^{k}\beta_j d_j^+(x_j), -\sum_{j=1}^{k}\beta_j d_j^-(x_j)\right)^T$ when n channels are assigned to the first k clients only. When $k = 0$, $\vec{f}(n,k) = \vec{0}$. Let $\vec{g}_k(n) = (d_j^+(n), -d_j^-(n))^T$ and $N' = \min\left\{\sum_{j=1}^{K} n_j, N\right\}$. $\vec{g}_k(n)$ is the vector of unsatisfactory function and bonus function of user j if we assign n channels to him/her. The recursive relation of $\vec{f}(n,k)$ is given by

$$\vec{f}(n,k) = \text{lexmin}\left\{\vec{f}(n-n', k-1) + \vec{g}_k(n') : 0 \le n' \le \min\{n_k, n\}\right\} \tag{4.5}$$

where $0 < n \le N'$ and $0 < k \le K$. By using this recursive relation, the assignment is obtained by means of evaluating $\vec{f}(N', K)$.

Proposition 4.1. *The inductive assignment algorithm provides the optimal assignment.*

The proof of proposition 4.1 is given in Appendix B.

In fact, the proof of proposition 4.1 is similar to the algorithm of obtaining $\vec{f}(N', K)$. The way to obtain the optimal $\vec{f}(N', K)$ and the corresponding channel assignment is outlined as follows. We can use a $N' \times (K+1)$ table to store each instance of $\vec{f}(n,k)$. The value of $\vec{f}(n,k)$ is stored in the n-th row and k-th column of the table.

Firstly, we put the zero vectors to column 0 because $\vec{f}(n,0) = \vec{0}$ for all n. Then, we evaluate the entries in column 1 by using the recursive relation in equation (4.5) and the column 0's information. Next, we compute the entries in column 2 in a similar way based on column 1's information. Then, we continue this process column by column until we reach the entry of $\vec{f}(N', K)$. By backtracking from column K, we can obtain the optimal number of channels to be assigned for each user.

In the calculation of each entry (n,k), we have to make use of the information of entries $(0,k-1), (1,k-1), \ldots, (n,k-1)$. There are totally $(K+1)N'$ entries to be evaluated. Therefore, the time complexity for the inductive assignment algorithm is $O(N'^2K)$.

4.1.3 Extension of Algorithm

As mentioned in Section 4.1.2, the time complexity for the inductive assignment algorithm is $O(N'^2K)$. However, when the system is large, the time required to implement the proposed algorithm for the whole system may be too large for practical use. For example, if we are assigning channels to more than one cells which are adjacent to one another, then the number of users involved can be very large.

This problem can be solved by a parallel processing implementation as follows. We can divide all the users into several groups. For example, if we are performing the dynamic channel assignment [29] for multiple cells which are adjacent to one another, we can divide the users according to the cell they reside.

We can compute the set of values of $\vec{f}(n,k)$ among the users in each group with the inductive assignment algorithm simultaneously. Then, we treat each group as a virtual user. Let $\vec{f}_j(n,k) = (f_j^{(1)}(n,k), -f_j^{(2)}(n,k))^T$ be the value of $\vec{f}(n,k)$ for the j-th group. Suppose K_j is the number of users in group j. The unsatisfactory function and the bonus function for each virtual user are

$$d_j^+(x_j) = f_j^{(1)}(x_j, K_j) \tag{4.6}$$

$$d_j^-(x_j) = f_j^{(2)}(x_j, K_j). \tag{4.7}$$

Now, the unsatisfactory function, $d_j^+(x_j)$, of group j is the minimum weighted sum of unsatisfactory function of all users in group j when we assign x_j channels to them. The bonus function, $d_j^-(x_j)$, of group j is the weighted sum of bonus function of all users in group j when we optimally assign x_j channels to them. As mentioned in section 4.1.2, we use a table to store every instances of $\vec{f}(n,k)$ in the implementation of the inductive assignment algorithm . From the last column of the table for the channel assignment of each group, we can obtain the function values in equations (4.6) and (4.7).

Then, we can apply the proposed algorithm again for those groups to complete the channel assignment. In this case, we consider each group as a virtual user with the unsatisfactory function and bonus function being (4.6) and (4.7). We perform the optimal channel assignments for these virtual users. We can then obtain the optimal number of channels assigned to each group. From the dynamic programming table of each group, we can backtrack the optimal number of channels assigned to each user eventually.

Since we compute the values for each user group in parallel first, the time complexity can be reduced although the amount of computation is the same. It can be easily proved that this parallel implementation can also provide the optimal solution for the larger system by induction in the same way as the proof for the inductive assignment algorithm.

In addition, if there are some new users joining the network, we can use similar method to adaptively assign the channels. Each new user is a group and all the old users are gathered into another group. Then, use the above method to assign the channels to all these 'groups' according to the old inductive assignment algorithm table of the network. In this case, when new users arrive at the system, we can have a more effective way to assign the channels.

4.2 Single Channel Assignment

We look into another special case in this section. In this section, we consider the scenario that at most one channel can be assigned to each user. That means, $n_j = 1$ for all j. This is the case for most of the current cellular system. Fortunately, as mentioned at the very beginning of this chapter, polynomial-time optimal algorithms are found for this case. Furthermore, we will adopt the optimal algorithm in Chapter 5 to obtain a lower bound of performance analysis. This is the reason why we consider this special case in this chapter.

4.2.1 System Model

Since for each user, at most one channel can be assigned to him/her, the achievement function in equation (3.15) can be converted as below:

$$\text{lexmin} \quad \left(\sum_{i=1}^{N} \sum_{j=1}^{K} d_{i,j}^{+} x_{i,j}, - \sum_{i=1}^{N} \sum_{j=1}^{K} d_{i,j}^{-} x_{i,j} \right)^{T} \tag{4.8}$$

where

$$d_{i,j}^{+} = \beta_j d_j^{+}(R_{i,j}), \tag{4.9}$$

$$d_{i,j}^{-} = \beta_j d_j^{-}(R_{i,j}). \tag{4.10}$$

Then, in constraint (3.16), we set each n_j to be 1.

However, there is a problem if we modify the model in this way. The new unsatisfactory function of the user becomes $\sum_{i=1}^{N} d_{i,j}^{+} x_{i,j}$.

This is not a monotonic decreasing function of $\sum_{i=1}^{N} R_{i,j} x_{i,j}$, which is a requirement in the choice of unsatisfactory function (see section 3.1). If $x_{i,j} = 0$ for all i and $d_{i,j}^{+}$ are positive for all i, the unsatisfactory function becomes 0 which is smaller than the case that there exists an $x_{i,j} = 1$. The new unsatisfactory function is not equal to the original unsatisfactory function when $x_{i,j} = 0$.

We can further modify the model as follows. We use a common technique which has been applied in many transportation problems and assignment problems [35]. If there are more channels than users, we add dummy users to the problem so that $N = K$. The unsatisfactory function and bonus function of these users are 0. Hence, $d_{i,j}^{+} = d_{i,j}^{-} = 0$ for all channels of these users. On the other hand, if there are more users than the channels, we add dummy channels to the problem so that $N = K$. Then, for each user, the values of $d_{i,j}^{+}$ and $d_{i,j}^{-}$ of the those dummy channels are $d_j^{+}(0)$

and $d_j^-(0)$ respectively. Finally, we use the following system constraints:

$$\sum_{i=1}^{N} x_{i,j} = 1, \quad \forall j \tag{4.11}$$

$$\sum_{j=1}^{N} x_{i,j} = 1, \quad \forall i \tag{4.12}$$

Now, both component of the achievement function are linear functions of $x_{i,j}$. Hence, we can now have a *linear binary goal programming* problem. That means, we can apply the *multiphase simplex algorithm* [12]. Alternatively, we can have the following two algorithms.

4.2.2 Proposed Algorithms

There are two more approaches to solve this problem. One is to extend the Hungarian Method [16]. Another one is to extend the linear programming algorithms.

Modified Hungarian Method

The problem formulation can be summarized as below:

$$\text{lexmin} \quad \left(\sum_{i=1}^{N} \sum_{j=1}^{K} \beta_j d_{i,j}^+ x_{i,j}, - \sum_{i=1}^{N} \sum_{j=1}^{K} \beta_j d_{i,j}^- x_{i,j} \right)^T \tag{4.13}$$

subject to

$$\sum_{i=1}^{N} x_{i,j} = 1 \quad \forall j \tag{4.14}$$

$$\sum_{j=1}^{N} x_{i,j} = 1 \quad \forall i \tag{4.15}$$

$$x_{i,j} \in \{0, 1\} \tag{4.16}$$

For each component of the achievement function, this formulation looks similar to the *assignment problem*[16] (see Appendix C for details) formulation except that we have a lexicographical minimum achievement function instead of a single minimization objective function.

To solve the problem, firstly, we only consider the minimization of weighted sum of unsatisfactory function in the achievement function (4.13). Then, we perform the Hungarian method (see Appendix C) to solve this single-objective problem. If there is only one combination from the modified cost matrix in such a way that the sum is zero, we can terminate according to the way of comparison in Definition 1.1.

Otherwise, we need to consider the maximization of the weighted sum of bonus functions. In the minimization of the weighted sum of unsatisfactory functions, there are zero entries in the modified cost matrix. Actually, these entries are the candidate assignments. Therefore, we constraint ourselves to these assignments in the maximization of the weighted sum of bonus functions. The way to do so is to construct another cost matrix based on the modified cost matrix in the previous steps and the values $\beta_j d^-_{i,j}$. Since we do not consider those nonzero entries in the modified cost matrix, in the new cost matrix, those corresponding entries are negative infinity. For other entries, in the new cost matrix, their values are $\beta_j d^-_{i,j}$.

Since assignment problems are minimization problems but we want to maximize the weighted sum of bonus functions, we need to change the cost matrix before implementing the Hungarian method. We only need to multiply each entry by -1 so that the Hungarian method will minimize the negative of the weighted sum of bonus functions which is equivalent to maximizing the weighted sum of bonus functions.

Linear Programming Approach

The second approach is to apply linear programming algorithms twice. As mentioned above, the problem can be viewed as two assignment problems. In fact, each assignment problem can be solved by linear programming algorithms directly due to the integer solution property [10]. We can solve the problem as follows.

Firstly, we replace the integer solution constraint with $0 \leq x_{i,j} \leq 1$. Then, we ignore the second component of the achievement function in (4.13). We only consider the weighted sum of unsatisfactory function. Next, we apply any linear programming algorithms to minimize the weighed sum of unsatisfactory function. This can be done by common mathematical software. After that, we add the following constraint to the problem:

$$\sum_{i=1}^{N}\sum_{j=1}^{N} d_{i,j}^{+} x_{i,j} = D_{\min}^{+} \tag{4.17}$$

where $D_{\min}^{+}$ is the minimum value of weighted sum of unsatisfactory function obtained in the previous steps. Now, we change the objective function to the weighted sum of bonus function. We treat it as another linear programming problem again and solve it by mathematical software. Then, we can obtain the final solution.

Chapter 5

Performance Evaluation

In this chapter, we will evaluate the performance of proposed algorithms in chapter 3 and 4 by means of computer simulations. We will compare the proposed algorithms with throughput optimization schemes. Since minimizing the weighted sum of unsatisfactory function is the most important objective in our channel assignment scheme, we will adopt it as a measure of performance.

5.1 General Channel Assignment and Single Channel Assignment

In this section, we would like to compare the performances of the throughput optimization scheme, two proposed algorithms in chapter 3, namely, the channel swapping algorithm and the best-first-assign algorithm, via simulations. We compare the performances by varying the number of clients and channels in the system. A lower bound of weighted sum of unsatisfactory function is provided for a reference. We will see later for the cases of $N \leq K$, $n_j = 1$, the lower bound of weighted sum of unsatisfactory function is obtained by the algorithm described in Section 4.2. Hence, we also compare this algorithm with other proposed channel assignment schemes and the throughput optimization scheme. In this simulation, we consider two sets of unsatisfactory function and bonus function.

5.1.1 System Model

We consider an MC-CDMA system with random orthogonal sequences. In this system, sequences are first generated randomly and then by using QR decomposition [15] or Gram-Schmidt procedure [17], the sequences become orthonormal. We assume large scale path loss is compensated by downlink power control methods [30]. The small scale fading is assumed to be Rayleigh fading and the background noise is assumed to be the additive white Gaussian noise (AWGN). We choose $R_{i,j}$ to be the throughput obtained by client j when channel i is assigned to him/her. In this simulation, $R_{i,j}$ is chosen to be the channel capacity of the channel. For each client j, he/she demands a throughput of D_j.

As mentioned above, we evaluate the performance of the proposed algorithms with two sets of unsatisfactory function and bonus function. The first set of unsatisfactory function and bonus function chosen are the equations below:

$$d_j^+\left(\sum_{i=1}^{N} R_{i,j}x_{i,j}\right) = \max\left\{0, D_j - \sum_{i=1}^{N} R_{i,j}x_{i,j}\right\} \tag{5.1}$$

$$d_j^-\left(\sum_{i=1}^{N} R_{i,j}x_{i,j}\right) = \max\left\{0, \sum_{i=1}^{N} R_{i,j}x_{i,j} - D_j\right\} \tag{5.2}$$

In this case, the value of unsatisfactory function means how much throughput is still needed for that client to meet the throughput demand. The value of the bonus function means how much throughput is beyond the client's minimum requirement. If the client's throughput is greater than or equal to this demanded throughput, the value of unsatisfactory function is zero. This is a typical data transfer scenario.

The second set of unsatisfactory function and bonus function chosen are the following equations:

$$d_j^+\left(\sum_{i=1}^{N} R_{i,j}x_{i,j}\right) = \frac{D_j(1+\text{sgn}(D_j - \sum_{i=1}^{N} R_{i,j}x_{i,j}))}{2} \tag{5.3}$$

$$d_j^-\left(\sum_{i=1}^{N} R_{i,j}x_{i,j}\right) = \max\left\{0, \sum_{i=1}^{N} R_{i,j}x_{i,j} - D_j\right\} \tag{5.4}$$

This is a typical choice of unsatisfactory function and bonus function for video streaming. The function $\text{sgn}(x)$ is the signum function where it is 1 if x is positive, 0 if x is 0 and -1 if x is negative. If the client's throughput is below the minimum requirement D_j, the unsatisfactory function is D_j because in this situation, no matter how high is the throughput, the buffer is starving [18] and it affects the playback of the video. Otherwise, it is 0. The bonus function is the same as the previous case.

For simplicity, we set $\beta_j = 1$ for all j which means the objective function of each user is equally important. The value of n_j is chosen to be $\lceil \frac{2N}{K} \rceil$ for $N \geq K$. Otherwise, we set $n_j = 1$.

In the performance evaluation in this section, we consider the *proportion of deficient throughput*, which is defined as follows, as the performance measure in our simulation.

Definition 5.1. *The* proportion of deficient throughput *of a channel assignment scheme is the weighted sum of unsatisfactory function this scheme divided by the weighted sum of unsatisfactory function without assigning any channels.*

Roughly speaking, this value is the proportion of the total client demand which has not been satisfied. In our choice of unsatisfactory function for this simulation, we are considering the proportion of throughput demand which has not been satisfied.

For the channel swapping algorithm, which is an iterative algorithm, we repeat the iterations until the solution cannot be further improved. That means, the algorithm terminates when the local optimal is reached. As mentioned in Section 3.4.1, the algorithm typically converges within 4 to 5 iterations in our simulations.

5.1.2 Lower Bound of Weighted Sum of Unsatisfactory Function

Apart from plotting the curves for the throughput optimization schemes, we also obtain a lower bound of the optimal weighted sum of unsatisfactory function as a benchmark for the performance evaluation. The lower bound is divided into two parts. The first part is for the case when $N > K$, i.e. $n_j > 1$. The second part is for the case when $N \leq K$, i.e. $n_j = 1$. Below is the description of how we obtain the lower bound.

When $N > K$, $n_j > 1$. The lower bound in this case is obtained by an algorithm which is likely to give an infeasible solution. In this algorithm, each client j independently obtains the n_j channels which have the highest throughput. Obviously, this gives the upper bound of the throughput of a user so this gives the lower bound of the unsatisfactory function of each client and thus the lower bound of weighted sum of unsatisfactory function. However, in this scheme, a channel may be assigned to more than one client so in general, the solution is infeasible as it violates the constraint (3.17). Hence, it is also a lower bound of the optimal weighted sum of unsatisfactory function.

When $N \leq K$, $n_j = 1$. In chapter 4, we have seen that for $n_j = 1$, there is a polynomial-time algorithm to give the optimal solution. Hence, we can make use of that algorithm to obtain the optimal solution as our lower bound. At the same time, we can compare the performance of that algorithm with other proposed algorithms and the throughput optimization scheme in this simulation.

Alternatively, since we only consider the weighted sum of unsatisfactory function, we can simply treat the problem as an *assignment problem* (see Appendix C). This is a single objective optimization problem and the objective function is a linear function. Although the decision variables are binary, by the integer solution property [10], we can simply relax the integer constraint and treat the decision variables as continuous variables from 0 to 1. Hence, we can obtain the lower bound for this case by the *Hungarian method* [16] or using some common mathematical software by formulating the problem as a linear programming problem.

5.1.3 Performance Evaluation I

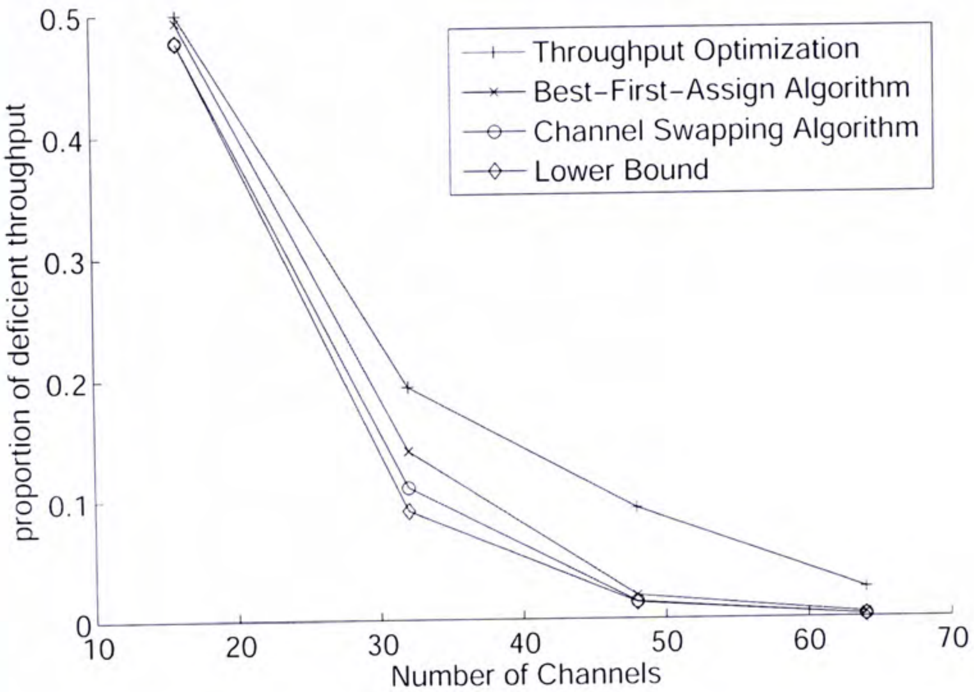

Figure 5.1: Relationship between the proportion of deficient throughput and number of channels

In this section, we present the results of the first test case. That is, the unsatisfactory function and bonus function are equations (5.1) and (5.2) respectively.

Firstly, we would like to compare how the performance varies with the number of channels. In this case, we fix the number of clients to be 16. As mentioned above, if $N \geq K$, we set $n_j = \lceil \frac{N}{K} \rceil$. Otherwise, we set $n_j = 1$. The results are plotted in figure 5.1. In this figure, as the number of channel increases, on average, more channels can be assigned to each user so the client requirements are easier to be met. Thus, it can be seen that all the curves decrease as the number of channels increases.

In addition, when the number of channels increase further, more and more users' throughput demands have been satisfied. That means, more and more users' unsatisfactory functions become zero which is the minimum possible value of unsatisfactory functions. In this case, fewer and fewer users' unsatisfactory functions can decrease further. Therefore, it can be seen that in figure 5.1, all the curves decrease with a smaller and smaller rate. This accounts for the shapes of the curve in figure 5.1.

In figure 5.1, it can be seen that the channel swapping algorithm and the best-first-assign algorithm perform better than the throughput optimization scheme. The curves for both of the channel swapping algorithm and the best-first-assign algorithm fall with higher rate than the one of the throughput optimization scheme. In addition, for $N > 48$, the curve of the channel swapping algorithm overlaps with our lower bound. That means, in those cases, the channel swapping algorithm converge to the optimal solution in those cases. This is a nice feature of the channel swapping algorithm.

What's more, it is observed that the curve of the best-first-assign algorithm is close to the one of channel swapping algorithm. This is an important feature for the choice of algorithms. When we consider the trade-off between computational complexity and the weighted sum of unsatisfactory function, this can give some hints on the choice of the algorithm. It will be discussed in more details in section 5.1.4.

Next, we would like to compare the variation of performance with respect to the number of clients. In this case, we fix the number of channels to be 16. Similar to the above case, we choose $n_j = \lceil \frac{N}{K} \rceil$ for $N \geq K$ and choose $n_j = 1$ for $N < K$. The results are plotted in figure 5.2. As the number of client increases, on average, fewer channels are assigned to each user so the proportion of deficient throughput becomes larger. Hence, it can be seen that those curves increase as the number of clients increase.

Moreover, it is observed that all the curves are concave which means the rate of increment is decreasing with the number of clients. When $K > 16$, since N is fixed to be 16, that means, some of the users do not have any channels. For these clients,

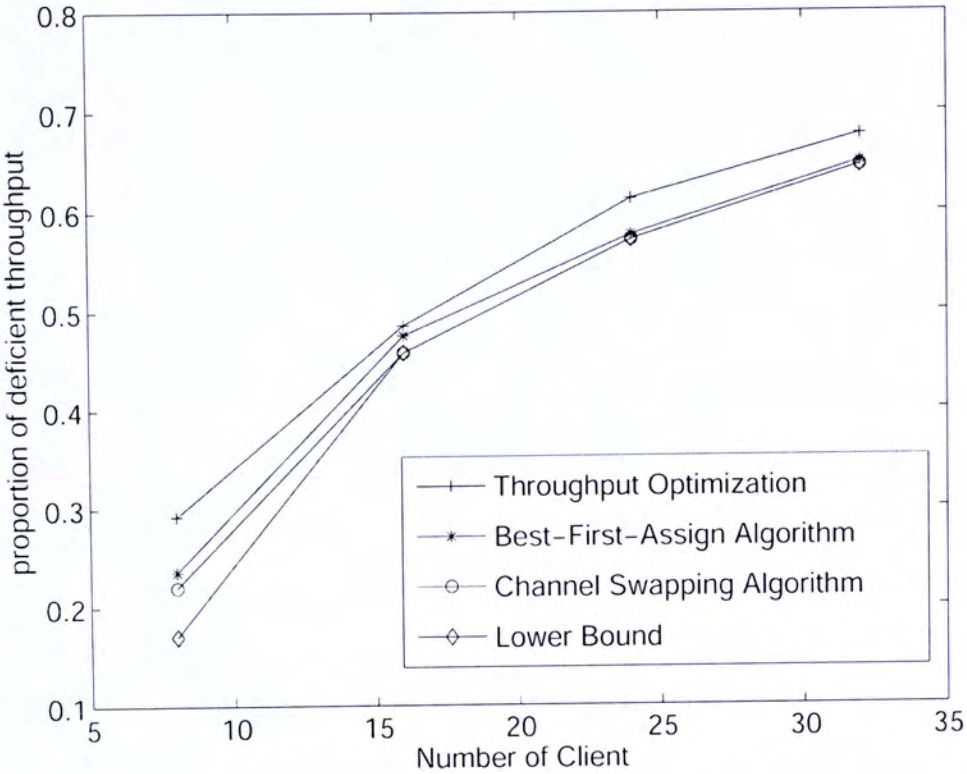

Figure 5.2: Relationship between the proportion of deficient throughput and number of clients

the unsatisfactory function is equal to their throughput demand D_j which is the maximum possible value for the unsatisfactory function. When the number of users increase, greater and greater proportion of the users who have the unsatisfactory function equal to their throughput demand. The rate of increment weighted sum of unsatisfactory function is closer and closer to the total throughput demand of the users. Hence, the proportion of throughput deficient, which is the weighted sum of unsatisfactory function divided by the total throughput demand of the users, increases with slower and slower speed. This accounts for the shape of the curves in figure 5.2.

In figure 5.2, it can be seen that the curves for the two proposed algorithms are lower than the one of the throughput optimization. When $K > 16$, it is more obvious that our proposed algorithms perform better. When $K > 16$, the channel swapping algorithm overlaps with the curve of the lower bound. That means, the channel swapping algorithm converges to the optimal solution. In addition, the curve of the best-first-assign algorithm gets closer and closer to the lower bound, especially when K is greater than 24.

5.1.4 Discussion

From the simulation results in figure 5.1 and figure 5.2, we found that the two curves of channel swapping algorithm and the best-first-assign algorithm are quite close to each other. That means, they have similar weighted sum of unsatisfactory functions. The channel swapping algorithm just performs a little bit better than the best-first-assign algorithm.

According to the simulation results, the channel swapping algorithm provides the lower weighted sum of unsatisfactory functions. In the above sections, it can be seen that the channel swapping algorithm converges to the optimal solution in large cases. However, the best-first-assign algorithm has much smaller time complexity. In section 3.4.1, the computational complexity is $O(N^3)$ for each iteration. On the other hand, in section 3.4.2, the computational complexity of the whole algorithm is $O(NK)$. This is an attractive feature of the best-first-assign algorithm. Therefore, for this set of unsatisfactory function and bonus function, unless we have to minimize the weighted sum of unsatisfactory function as much as possible, the best-first-assign algorithm is a better choice from the complexity point of view.

5.1.5 Performance Evaluation II

We consider the second set of unsatisfactory function and bonus function in this section. They are equations (5.3) and (5.4) respectively.

To begin with, we consider how the performance of the algorithms changes with the number of channels. Just like the first case, we fix the number of clients to be

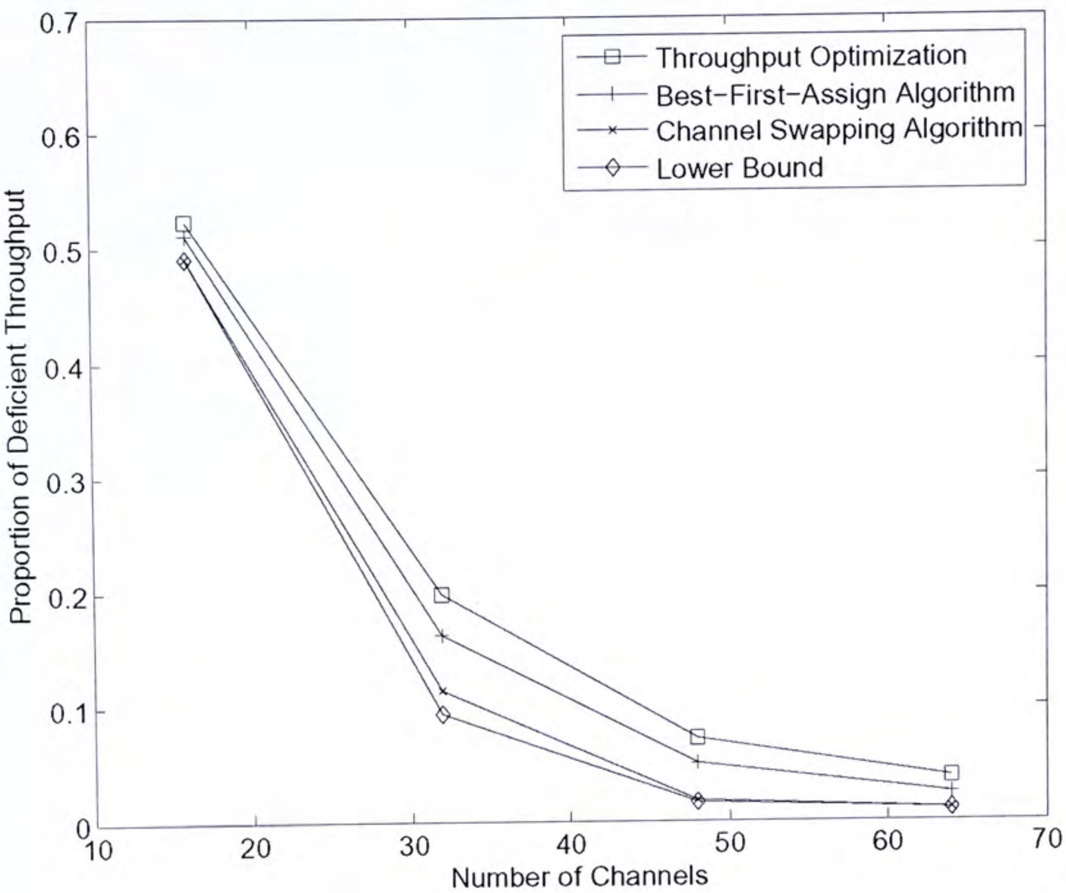

Figure 5.3: Relationship between the proportion of deficient throughput and number of channels

16. If $N \geq K$, $n_j = \lceil \frac{N}{K} \rceil$. Otherwise, $n_j = 1$.

Figure 5.3 shows the results. In this figure, all the curves drop as the number of the channel increases. Both the channel-swapping algorithm and best-first-assign algorithm performs better than the throughput optimization scheme. The curve of channel swapping algorithm is very close to the lower bound of the proportion of deficient throughput.

However, the curve of the best-first-assign algorithm is not so close to the one of the channel swapping algorithm as before. This is due to the fact that the unsatisfactory function is not a continuous function but it varies abruptly. If the throughput demand of the users are so high that none of them can be satisfied by a single channel, then in that particular iteration, the best-first-assign algorithm will decide the channel assignment based on the bonus function which is again the same for all users. In this case, it is similar to arbitrary channel assignment. Therefore, it does not perform well in this case. That means, we should choose the channel swapping algorithm for this kind of unsatisfactory function which has abrupt changes.

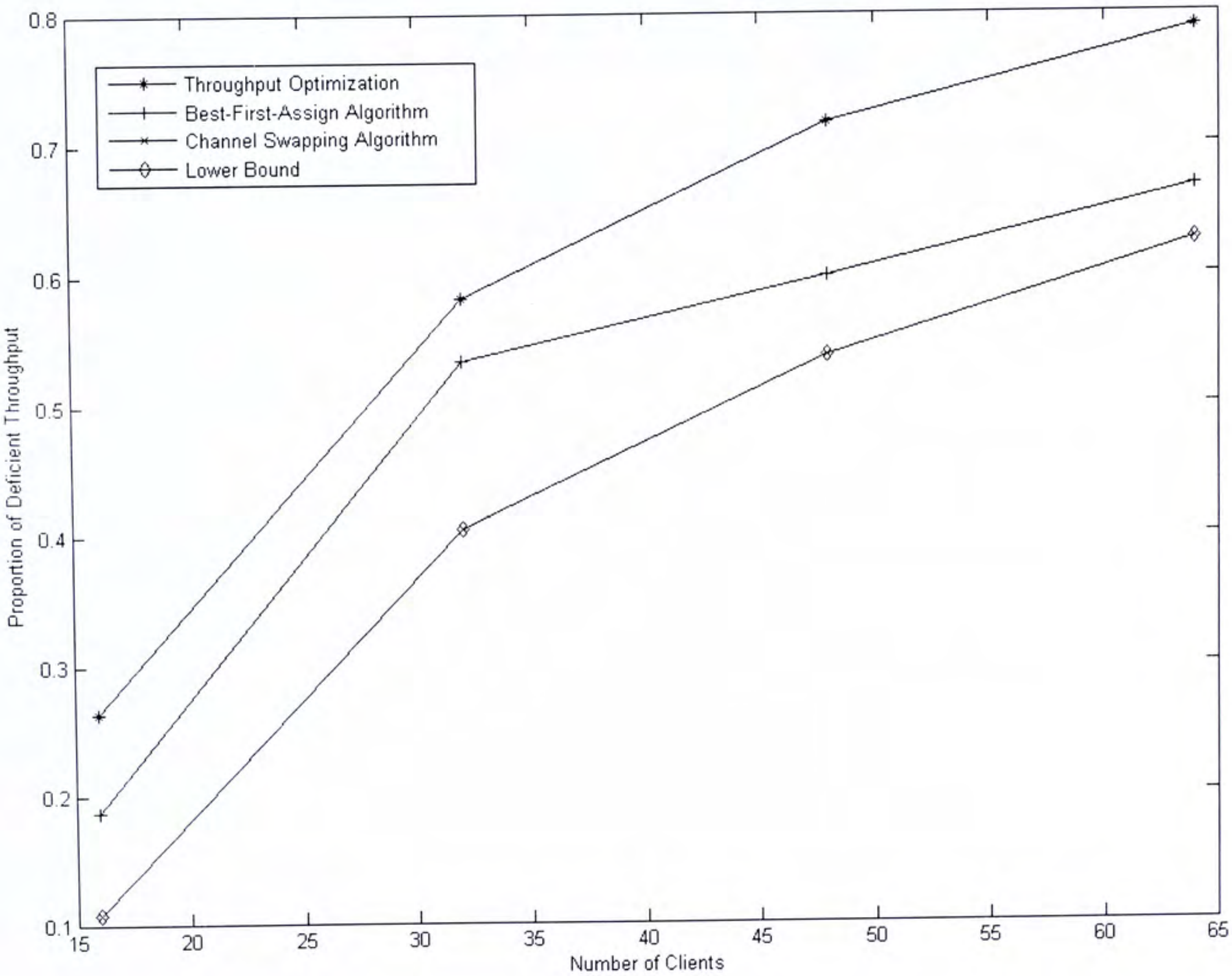

Figure 5.4: Relationship between the proportion of deficient throughput and number of clients

Then, we move on to investigate the variation of the performance of the algorithms with respect to the number of users. The results are shown in figure 5.4. In this case, we can also see that our proposed algorithms perform better than the throughput optimization algorithm. In this case, the curve of the channel swapping algorithm touches the lower bound of proportion of deficient throughput. That means, the channel swapping algorithm reaches the optimal solution. When the number of users is greater than 32, the curve of the best-first-assign algorithm becomes closer and closer to the lower bound but this does not happen for the throughput optimization scheme.

5.2 Single Order of Selection Diversity Algorithm

In this section, we would like to evaluate the performance of the proposed algorithm for the special case of order of section diversity being 1, which is described in chapter 4. The performance is evaluated via simulations. In our simulations, we would like to compare how the performances of that proposed algorithm and a throughput optimization scheme vary with number of clients and number of channels. In this section, we will also show the performance of the proposed algorithm and throughput optimization scheme in terms of weighted sum of bonus functions. Unlike the general case considered in the previous sections, when the order of selection diversity is 1, from the curves of the weighted sum of bonus functions, we can see some nice features of the proposed algorithm. This is the reason why although the weighted sum of the bonus function is not the most important objective function under our formulation, we also show the simulation results of the weighted sum of bonus function in this section. Just like in the performance evaluation of the general case algorithms, in this section, we will also evaluate the proposed algorithm by two similar sets of unsatisfactory function and bonus function.

5.2.1 System Model

Two scenarios of a narrowband DS-CDMA system with Hadamard signature sequences are considered. The first one is that the number of channel is fixed to be 16 while the number of clients varies. The second one is that the number of clients is fixed to be 16 while the number of channels varies.

We consider two sets of unsatisfactory function and bonus function. In the first simulation, the unsatisfactory function and bonus function of the clients are chosen to be:

$$d_j^+(x_j) = \max\{0, D_j - R_j x_j\} \tag{5.5}$$

$$d_j^-(x_j) = \max\{0, R_j x_j - D_j\} \tag{5.6}$$

where R_j is the throughput of each channel of client j and D_j is the demanded throughput of client j. The unsatisfactory function is the amount of throughput demand which cannot be satisfied. The bonus function is the amount of throughput beyond the minimum requirement. This simulation is used to imitate a scenario of typical data transfer.

In the second simulation, we choose the following unsatisfactory function and bonus function:

$$d_j^+(x_j) = \frac{D_j[1 + \operatorname{sgn}(D_j - R_j x_j)]}{2} \tag{5.7}$$

$$d_j^-(x_j) = \max\{0, R_j x_j - D_j\} \tag{5.8}$$

This simulation aims to imitate a scenario of video streaming. Since no matter how high the throughput is, if the amount of throughput is below the minimum requirement, the buffer starves and it affects the performance. Therefore, the unsatisfactory function is D_j if the total throughput is lower than D_j. The bonus function is the same as the one appeared in the first case.

The system is assumed to be synchronous. Match filter receiver is used to demodulate the signal. Since the Hadamard signature sequences are orthogonal and we consider a narrowband DS-CDMA system, the MAI is zero. The large scale path loss is compensated by perfect power control methods. The channel is assumed to be a slow Rayleigh fading channel. The background noise of the channel is assumed to be the additive white Gaussian noise (AWGN). In our simulations, the mean and standard deviation of the background noise are 0 and 10^{-4} respectively.

We adopt Hadamard signature sequence because the signal to noise ratio for each channel of the same user is shown to be the same [8]. The throughput of the channel is chosen to be the channel capacity of the channel.

In this simulation, the throughput of the channel is obtained by the above equation. The value of B is chosen to be 10 and D_j is uniformly distributed from 0 to 10000 inclusively. n_j is chosen to be $\lceil \frac{2N}{K} \rceil$. When $K > 2N$, n_j is chosen to be 1.

5.2.2 Performance Evaluation I

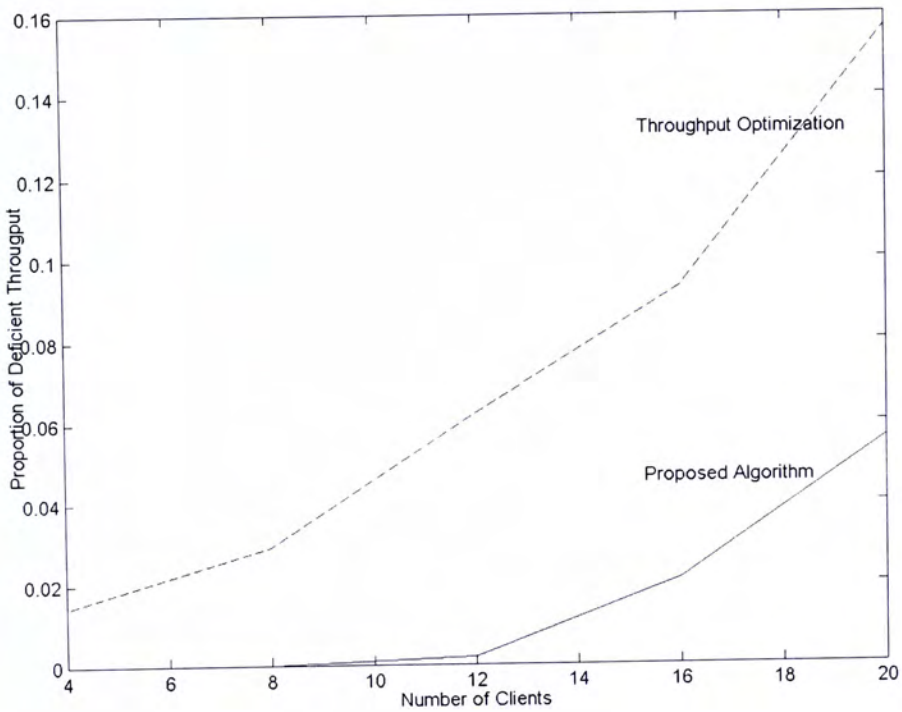

Figure 5.5: Relationship between proportion of deficient throughput and number of clients for fixed N

In this part, we consider the first set of unsatisfactory function and bonus function. They are equations (5.5) and (5.6) respectively.

In the first scenario, we fix the number of channels to be 16 and change the number of clients. We then plot the weighted sum of unsatisfactory function and bonus function against the number of clients in figure 5.5 and figure 5.6 respectively.

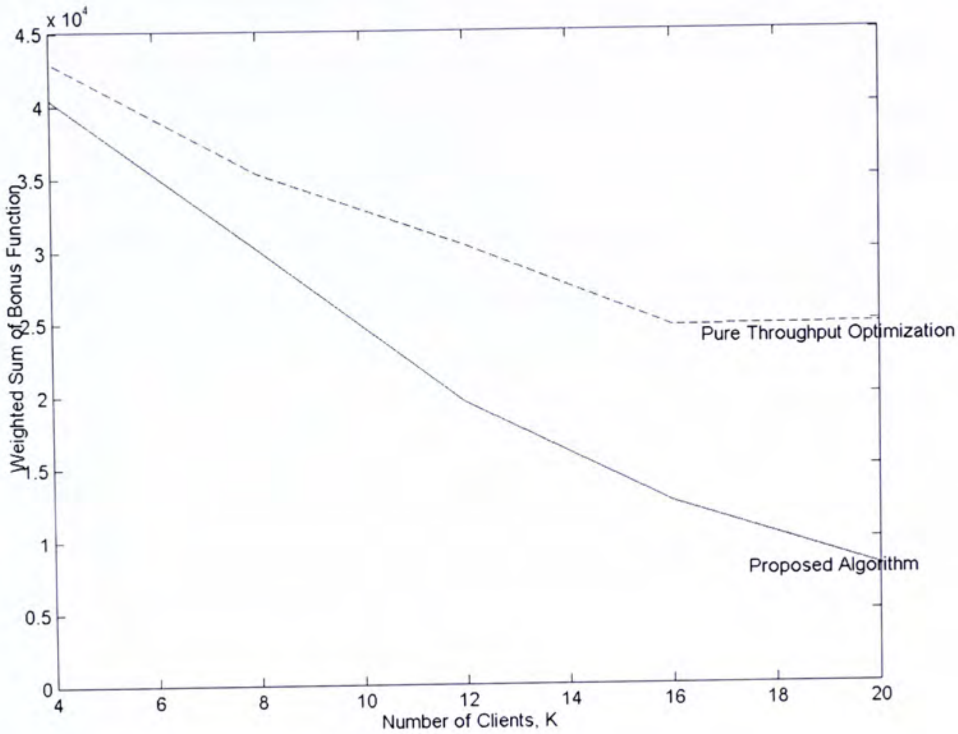

Figure 5.6: Relationship between weighted sum of bonus function and number of clients for fixed N

Based on figure 5.5, the curve for the weighted sum of unsatisfactory function is much lower for the proposed algorithm as expected (see chapter 4 and appendix B). It is initially at zero and starts to escalate when K is 8. On the other hand, for the throughput optimization scheme, the curve is already non-zero when K is 4. Since the number of channels is fixed, when there are more and more users, on average, each user can have fewer and fewer channels. Therefore, it is expected that both curves should increase as shown in figure 5.5. From this figure, it can be seen that the client unsatisfactory function increases more slowly.

In figure 5.5, the curve for the proposed algorithm is lower than the one for the throughput optimization. It is because in throughput optimization, more channels are assigned to the users whose channels, on average, have higher throughput than others. This contributes a lot in the weighted sum of bonus function of the throughput optimization scheme. On the contrary, our proposed algorithm tries to meet the client requirement as much as possible. Therefore, when compared with the throughput optimization scheme, our proposed algorithm provides fewer channels to those users whose channels, on average, have higher throughput than others. This explain why the weighted sum of bonus function of the proposed algorithm is smaller than the one of the throughput optimization scheme. This implies that the proposed algorithm is fairer than the throughput optimization scheme from the QoS point of view. It is because in our proposed algorithm, the channels are assigned so as to meet the client requirements as much as possible while the throughput optimization scheme tends to assign channels to those users who have had satisfactory performance already.

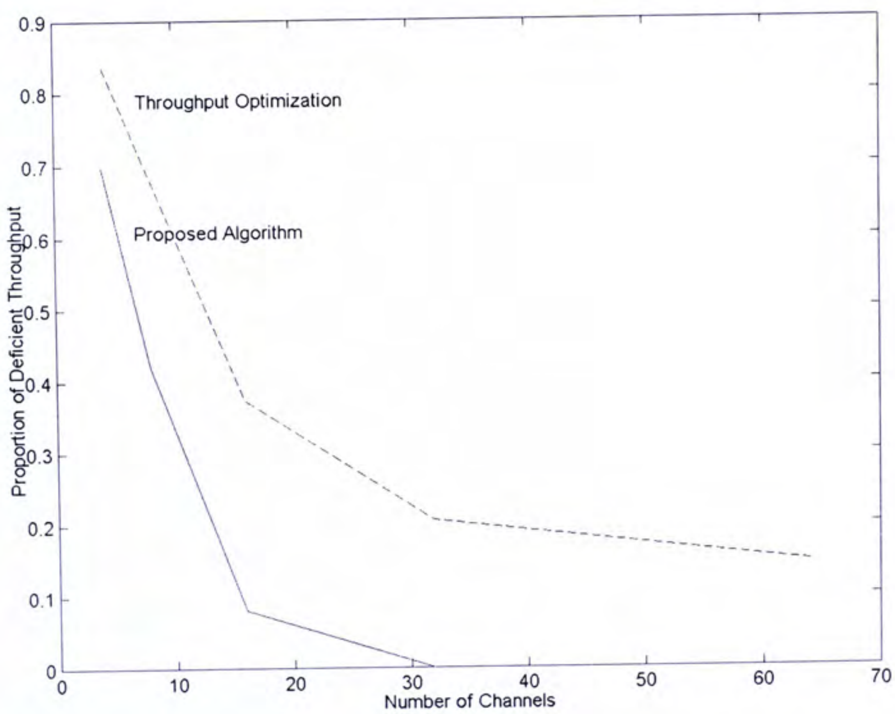

Figure 5.7: Relationship between proportion of deficient throughput and number of channels for fixed K

We plot the similar graphs (figure 5.7 and figure 5.8) for the second scenario, i.e.

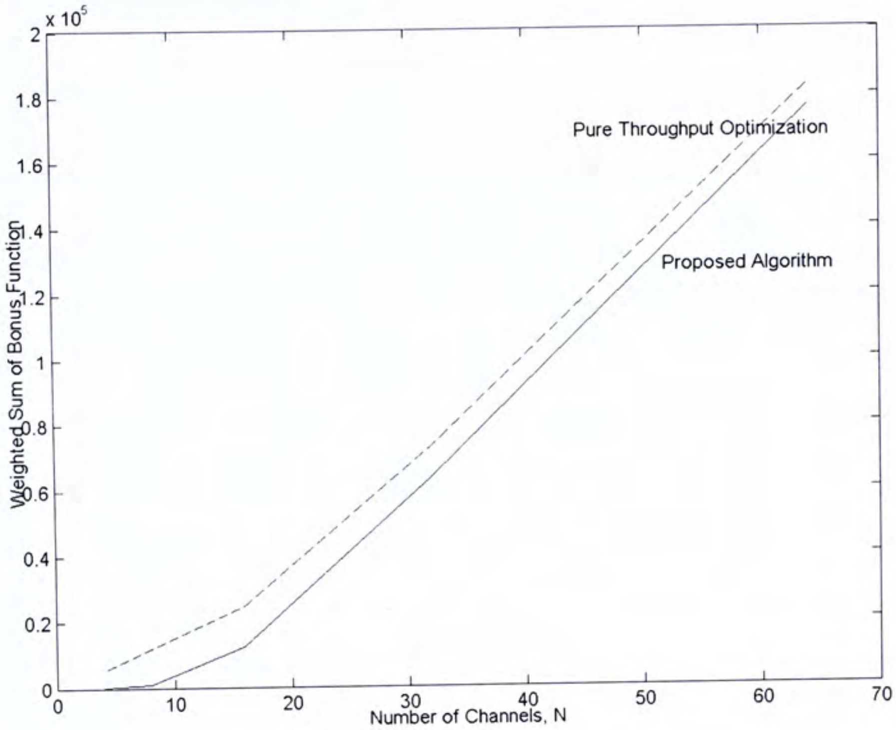

Figure 5.8: Relationship between weighted sum of bonus function and number of channels for fixed K

we fix the number of users and investigate how the weighted sum of unsatisfactory function varies with the number of channels. As the number of channels increases, on average, more channels can be assigned to each user. In this case, it is easier to satisfy each user's requirement. Therefore, it can be seen that both curves in figure 5.7 are decreasing while the curves in figure 5.8 are increasing as the number of channels increases.

In figure 5.7, it can be seen that both curves drop with a decreasing rate. The reason is as follows. When there are more channels, there are more users whose throughput requirement is completely satisfied which makes their unsatisfactory function becomes zero. Since the unsatisfactory function cannot be smaller than zero, when there are more channels, fewer users' unsatisfactory function can be decreased if we further increase the number of channels. Thus, the curves fall with a decreasing rate. This accounts for the convex shape for both curves shown in figure 5.7.

From figure 5.7, we can see that weighted sum of unsatisfactory function is smaller for the proposed algorithm. The curve for the proposed algorithm falls much faster than the one of throughput optimization. When N is 16, the weighted sum of unsatisfactory function of the proposed algorithm is less than one third of the one for the throughput optimization. When $N \geq 32$, the weighted sum of unsatisfactory function drops to zero. That means, our proposed algorithm only requires 32 channels to satisfy all the users' requirements. However, on the other hand, for the pure throughput optimization, the weighted sum of unsatisfactory function is still decreasing slowly. It is still far more than 0 even when N is equal to 64 which is twice the number of channels required for our proposed algorithm to meet the users' requirements. That means, our proposed algorithm assigns the channels in a more efficient way.

Furthermore, from figure 5.8, it can be seen that the weighted sum of bonus functions of the proposed algorithm is just slightly less than the one of the pure throughput optimization. Moreover, the slopes of both curves are nearly the same. This is an attractive feature for the proposed algorithm. This does not only show that the proposed algorithm requires fewer resources to meet the clients' demands, but also, on average, the performance of each user is close to the one for the pure throughput optimization.

5.2.3 Performance Evaluation II

In this section, we consider the second set of unsatisfactory function and bonus function which are equations (5.7) and (5.8) respectively.

Firstly, we fix the number of channels to be 16 and investigate how the performance of the proposed algorithm and the throughput optimization scheme varies as the number of clients increase.

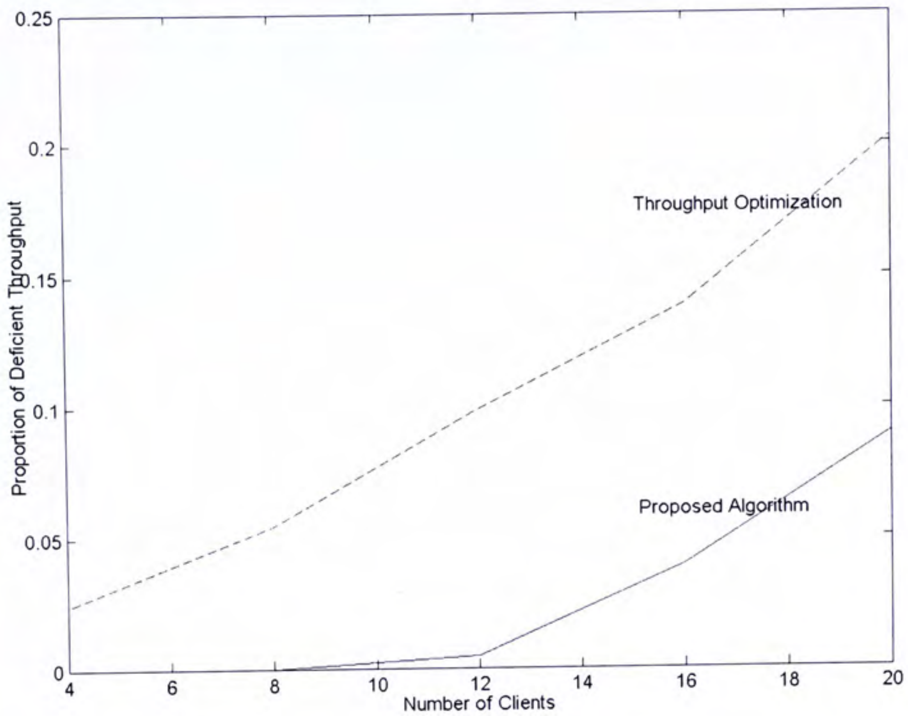

Figure 5.9: Relationship between proportion of deficient throughput and number of clients for fixed N

The results are plotted in figure 5.9 and figure 5.10. In figure 5.9, our proposed algorithm performs better than the throughput optimization scheme. The proportion of deficient throughput for the proposed algorithm is about 50% lower than the one of throughput optimization scheme.

In figure 5.10, the bonus function for throughput optimization scheme is much higher than the one of the proposed algorithm. The gap between these two curves increase as the number of clients increase.

From figure 5.9, we find that, on average, if we adopt the proposed algorithm, the throughput of each user is closer to the minimum requirement. On the other hand, if we apply the throughput optimization scheme, there are more users whose throughput is far below the minimum requirement. Figure 5.10 shows the average performance for those users whose throughput requirements are satisfied. It shows

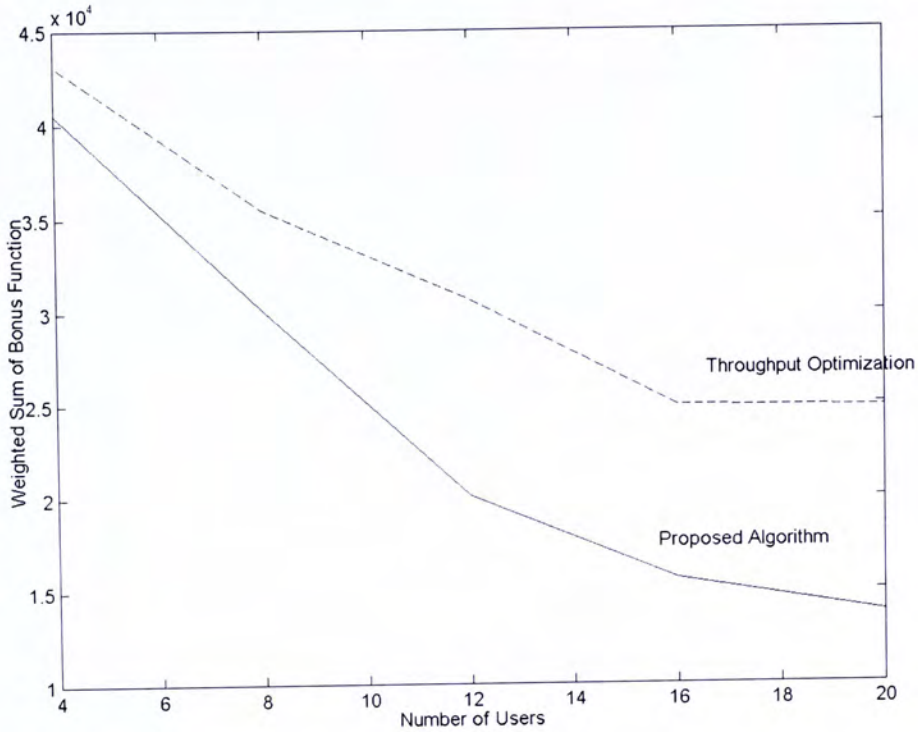

Figure 5.10: Relationship between weighted sum of bonus function and number of clients for fixed N

that for the throughput optimization, the average throughput of these users are much higher. However, just as what we have seen in figure 5.9, it tends to be more users whose throughput is far below the minimum requirement for throughput optimization scheme. That means, for throughput optimization, some users whose performance is far beyond the minimum requirement but there are also some users whose performance is far below the minimum requirement simultaneously. That means, compared to the throughput optimization scheme, the proposed algorithm is fairer.

Now, we move on to investigate how the performance of the proposed algorithm and throughput optimization scheme changes when we change the number of channels. We fix the number of users to be 16. The results are illustrated in figure 5.11 and 5.12.

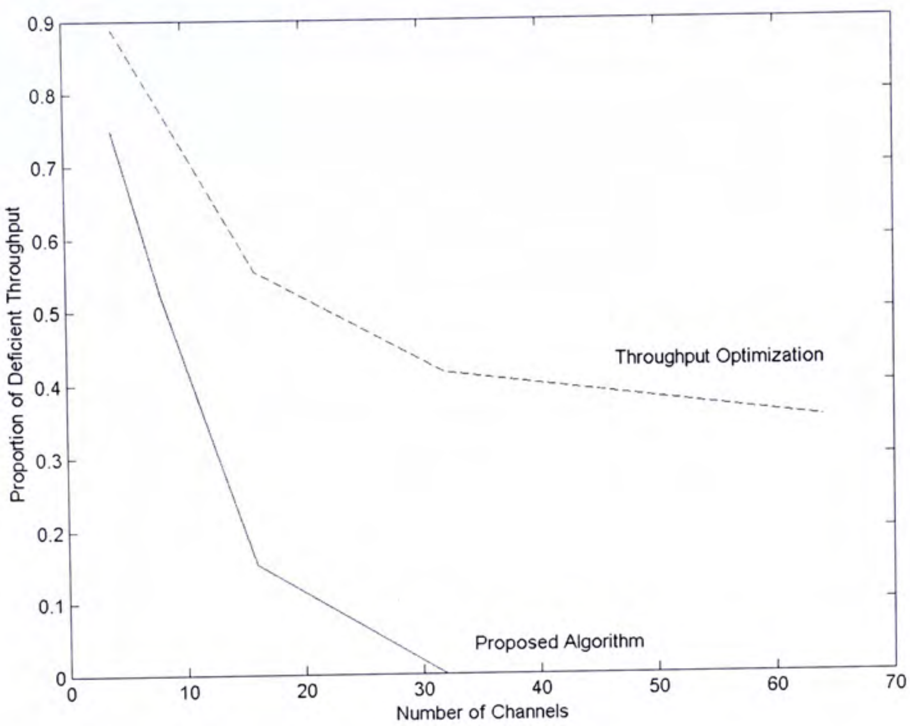

Figure 5.11: Relationship between proportion of deficient throughput and number of channels for fixed K

In figure 5.11, it can be seen that the curve for the proposed algorithm is fall below the one of the throughput optimization. The former drops much faster than the latter. When $N \geq 32$, the proportion of deficient throughput reaches 0 for the proposed algorithm. However, for the throughput optimization, it is still decreasing slowly and far from the x-axis.

However, in figure 5.12, we find that both curves are very close. That means, for those users whose throughput requirements are met, their throughput is quite similar for both the proposed algorithm and throughput optimization scheme. This means that the proposed algorithm does not only meet more throughput demands, but also achieves very satisfactory amount of throughput.

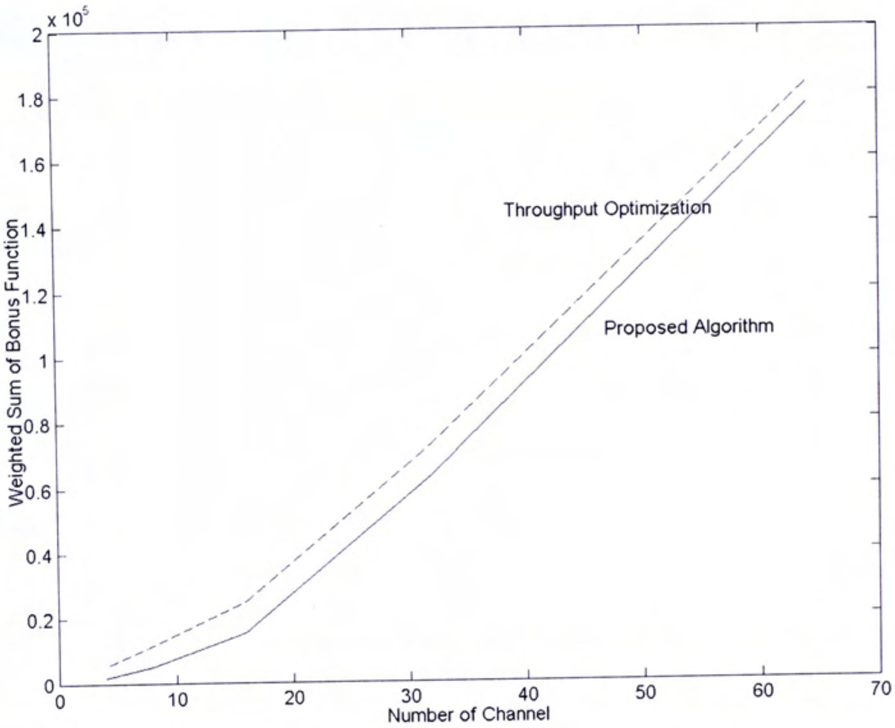

Figure 5.12: Relationship between weighted sum of bonus function and number of channels for fixed K

Chapter 6

Conclusion and Future Works

6.1 Conclusion

In this thesis, we focus on applying goal programming technique in channel assignment problem. Firstly, we formulate the goal programming model for the general channel assignment problem and then we propose two corresponding channel assignment schemes. Secondly, we consider two special cases of the channel assignment schemes. The goal programming models for these two special cases are modified and new optimal polynomial-time channel assignment schemes are proposed. Simulations have been performed to compare our proposed schemes with the throughput optimization scheme. Based on the simulations results, we compare the two channel assignment schemes of the general case.

To begin with, we consider the general channel assignment problem. In this model, we do not assume the underlying multiple access scheme except we only require each logical channel should be orthogonal to one another, i.e. the MAI is 0. The property of channel for the user application is specified by the *quality index*. The QoS requirements of each client are specified by a pair of functions, namely, the *unsatisfactory function* and the *bonus function*. Each user has two objective functions (minimizing the unsatisfactory function and maximizing the bonus function) and all these objective functions conflict with one another in general. Goal programming technique is then applied to tackle this problem and then we have the goal programming formulation for the general channel assignment problem.

It is shown that the problem is an NP-hard problem. Therefore, we propose two near-optimal polynomial-time channel assignment schemes, namely the *channel-swapping algorithm* and the *best-first-assign algorithm*. The channel-swapping algorithm has slightly better values for the achievement function while the best-first-assign algorithm has much smaller time complexity (although the channel-swapping algorithm usually converges within 4 to 5 iterations). However, according to the simulation results, the weighted sum of unsatisfactory function provided by both algorithms are very close. Therefore, the choice of algorithm is concluded as follows. Unless the weighted sum of unsatisfactory function is much more important than the time complexity, the best-first-assign algorithm is preferred.

Our work does not end here. Since the algorithms mentioned above are only near-optimal algorithms and the general case is an NP-hard problem, we then move on to seek for optimal algorithms in some special cases which are typical in communication systems. In the first special case, the *order of selection diversity* of all the users is equal to 1. In the second case, we consider the case of single channel assignment (i.e. we assign at most 1 channel to each user).

In the first special case, since the order of selection diversity of the system is 1, that means for each user, the channels are homogenous. However, for the same channel, the performance for different client is different in general. The problem is then reduced to how many channels should be assigned to each user. A dynamic programming algorithm is proposed to provide the optimal solution. It is also noted that this algorithm is not only applicable to single cell systems, but also can be extended to multi-cell systems in which each cell is adjacent to one another. Parallel processing implementation of this extension is also described to reduce the time complexity of the whole assignment. In addition, it is found that when there are new users requesting for channels, we can adaptively perform the assignment. From the simulation results, it found that the proposed algorithm does not only have lower weighted sum of unsatisfactory function, but also in some cases, the weighted sum of bonus function is very close to the throughput optimization. This is an attractive feature of the proposed algorithm.

In the second special case, each user can be assigned to at most 1 channels which is a very common scenario in present communication systems. It is also useful in obtaining the lower bound of the weighted sum of unsatisfactory function for the performance evaluation in the general case.

The problem is transformed into two assignment problems and we can solve it with two approaches. Firstly, as assignment problems can be solved by linear programming algorithms by relaxing the integer solution constraint, we can use some common mathematical software to solve the assignment problems one-by-one by adding a constraint when solving the second problem.

Alternatively, assignment problems can also be solved by Hungarian methods. Therefore, we can apply the Hungarian methods twice to obtain the optimal solution with two different cost matrices.

6.2 Future Works

The goal programming approach provides a new outlook in the channel assignment problem. There are still a lot of problems that require our effort to investigate. In this section, some suggested research directions are provided.

6.2.1 Multi-cell Channel Assignment

In this thesis, most of the systems that we consider are single cell systems. In practice, operators concern more about the multi-cell channel assignment. In multi-cell channel assignment scenario, if all the cells are adjacent to one another, we can consider the whole system as a larger cell and directly apply the proposed algorithms in this thesis. However, in most of the cases, channel reuse [29] is possible as the operators have a large number of cells covering a large area. A channel can be assigned to more than one users if they are far enough to have sufficiently small co-channel interference. To model this problem, we cannot impose constraint (3.17). A more general model and scheme may be needed.

There are two main types of channel assignment schemes for multi-cell channel assignment [29]. The first type is known as *fixed channel assignment* (FCA). In FCA, a predetermined set of channels is assigned to each cell. Thus, we can directly apply the proposed algorithms for each cell to have independent channel assignments.

However, the main drawback of FCA is lack of flexibility. Users can only be assigned to the available channels in their resided cell. In this case, some channels are not used while in some other cells, the users are blocked. Resource is wasted in this case. To solve this problem, the second type of multi-cell channel assignment scheme is introduced, which is known as *dynamic channel assignment* (DCA) [14].

In DCA, when a channel request is made, the serving base station of the cell requests for a channel from the mobile switching center (MSC). The switching center then allocates a channel to the requested cell according to a certain channel assignment algorithm. In this case, the channel assignment scheme is more flexible and efficient. The blocking probability can also be greatly reduced.

In DCA, a channel assignment scheme called *sequential packing* [34] is used in traditional voice network. However, this algorithm only takes care of number of requested channels in each cell but ignores the QoS requirements of each user. But for data network, QoS requirement plays an important role in resource allocation. That means, we need to have a new channel assignment scheme.

The challenge of multi-cell channel assignment is that the number of users and channels involved is much larger. In the single cell case, the problem is NP-hard already. That means, the multi-cell channel assignment problem is much harder to be solved as the problem size is greater. This is the main challenge of this problem.

6.2.2 Theoretical Studies

Another research direction is to obtain some theoretical limits of the amount of resource required to provide certain grade of service. This can give the operators a picture of how much resource they need to invest to provide a particular grade of service. Examples are the Erlang B or C formulas in traditional cellular network which only provides voice services. In addition, if the operators would like to upgrade their present services, they also need to know how much the minimum additional cost they need to pay to decide whether it is worth to have this investment. It is similar to the sensitivity analysis in some operations research problems.

In chapter 4, we proposed the optimal algorithms for two special cases. From the operator point of view, the next question is how many channels are required so that we can support certain user requirements based on the proposed algorithm. This depends on the traffic models and the specifications of the user applications (i.e. the unsatisfactory function and bonus function).

For the general case, the optimal polynomial-time algorithm is hard to obtain as it is an NP-hard problem. In this case, one possible direction of research is to obtain some bounds of the number of channels required to meet the user requirements.

6.2.3 Adaptive Algorithms

All the proposed algorithms in this thesis are not adaptive algorithms. They all need to perform the channel assignment in each time slot independently. They never make use of the results in the previous time slots and the relationship between the problems in consecutive time slots.

In practical situations, the channel conditions between two consecutive time slots do not vary too much if the time between two consecutive time slots is smaller than the coherence time of the channel. In addition, it is expected that the number of users and their user requirements seldom have drastic changes in the next time slot. We can have faster implementations by making good use of the correlation between the consecutive time slots. To reduce the time complexity, we may perform the channel assignment based on the results in the previous time slots and the change in the current time slot. This is the motivation of investigating adaptive algorithms for the channel assignment problem.

6.2.4 Assignment of Non-orthogonal Channels

In the formulation of this thesis, we assume that the channels are orthogonal to each other. That means, the MAI of the channels is zero. However, this assumption limits the number of the channels that can be assigned.

There are many research works in multiple access schemes which have non-orthogonal channels but they have satisfactory signal-to-interference ratio. One example is the introduction of carrier interferometry (CI) signal in multiple access schemes which have orthogonal channels [1][22][23][41]. With these techniques, we can have more channels to be assigned to the users.

Hence, a possible research direction is in the channel assignment for multiple access schemes with non-orthogonal channels. Besides the multi-cell channel assignment, this is another direction of generalization of channel assignment problem. However, in this case, we do not only need to consider the assignment of channels, but also other parameters like transmitted power in each channel because the MAI in these cases is no longer zero. Some other techniques like downlink power control [13] and multiuser detection [37] may be applied jointly as well.

Appendix A

Proof of Proposition 3.1

Proposition 3.1 is quoted below:

Proposition 3.1. *The optimization problem formulated in equations (3.15) to (3.18) is an NP-hard problem .*

Consider this special case. Assume there are N channels and 2 clients who can support at most N channels. Suppose $R_{i,1} = R_{i,2}$ are all nonnegative integers which correspond to the channel throughput. Let

$$d_1^+(\sum_{i=1}^{N} R_{i,1}x_{i,1}) = \max\left\{\frac{\sum_{i=1}^{N} R_{i,1}}{2} - \sum_{i=1}^{N} R_{i,1}x_{i,1}, 0\right\} \tag{A.1}$$

$$d_1^-(\sum_{i=1}^{N} R_{i,1}x_{i,1}) = 0 \tag{A.2}$$

$$d_2^+(\sum_{i=1}^{N} R_{i,2}x_{i,2}) = 0 \tag{A.3}$$

$$d_2^-(\sum_{i=1}^{N} R_{i,2}x_{i,2}) = 2\sum_{i=1}^{N} R_{i,2}x_{i,2} - \sum_{i=1}^{N} R_{i,1}. \tag{A.4}$$

In this case, client 1 demands half of the total throughput of all channels while client 2 does not have a demanded throughput. From equation (A.1) to equation (A.4), it can be seen that d_1^+ and d_2^+ are monotonic decreasing functions of $\sum_{i=1}^{N} R_{i,1}x_{i,1}$ and $\sum_{i=1}^{N} R_{i,2}x_{i,2}$ respectively. Also, d_1^- and d_2^- are monotonic increasing functions of $\sum_{i=1}^{N} R_{i,1}x_{i,1}$ and $\sum_{i=1}^{N} R_{i,2}x_{i,2}$. Hence, d_1^+ and d_2^+ are valid unsatisfactory functions and d_1^- and d_2^- are valid bonus functions.

Proposition A.1. *This special case problem is an NP-complete problem.*

Proof. We will show that this is the *number partitioning problem* which has been shown to be NP-complete [9]. In the number partitioning problem, a set of N non-negative integers, S, is given. The objective is to divide this set into two partitions such that the sum of these two sets are as close as possible.

Let $R_{i,1}$ (and hence $R_{i,2}$) be the i-th element in this set. Without loss of generality, we do the partition such that the first partition, P_1, has larger sum than another partition, P_2. $x_{i,1} = 1$ if the i-th element of S is assigned to P_1. Otherwise, $x_{i,2} = 1$.It can be easily seen that in our special case problem, $x_{i,1} + x_{i,2}$ must be equal to 1 so as to either minimize d_1^+ (by setting $x_{i,1} = 1$) or maximize d_2^- (by setting $x_{i,2} = 1$).

Since P_1 has larger sum, $\sum_{i=1}^{N} R_{i,1}x_{i,1} \geq \frac{\sum_{i=1}^{N} R_{i,1}}{2}$. Hence, d_1^+ is always zero.

The difference between sum of P_1 and P_2 is given by

$$\sum_{i=1}^{N} R_{i,1}x_{i,1} - \sum_{i=1}^{N} R_{i,2}x_{i,2} = \sum_{i=1}^{N} R_{i,1} - 2\sum_{i=1}^{N} R_{i,2}x_{i,2}. \tag{A.5}$$

To minimize this difference is equivalent to maximize the negative of it, i.e. d_2^- in our special case problem. Thus, the optimal solution of the special case problem is exactly the solution of the number partitioning problem which has been proved to be NP-complete. Hence, this special case problem is an NP-complete problem. □

Since the above problem is an NP-complete problem and it is a special case of our channel assignment problem, we can have the following corollary.

Corollary A.1. (Proposition 3.1) *The general channel assignment problem is an NP-hard problem.*

Appendix B

Proof of Proposition 4.1

Proposition 4.1 is quoted below:

Proposition 4.1. *The inductive assignment algorithm in Chapter 4 provides the optimal assignment.*

In this chapter, I will provide the proof of Proposition 4.1. It shows that the inductive assignment algorithm in Section 4.1.2 is an optimal algorithm for the case of single order of selection diversity.

Firstly, we prove that the recursive relation of $\vec{f}(n,k)$ is true for all n and k by induction. If it is true, from the meaning of this function, the inductive assignment algorithm must give the optimal solution. It is obviously true for both n and k are 0. Also, when $k = 0$, there are zero terms in the summations of both components in equation (4.1). Therefore, it is also true for all values of n when $k = 0$.

Assume it is true for $0 \leq n \leq n'$ and $k = k'$ where n' and k' are positive integers such that $0 \leq n' \leq N'$ and $0 \leq k' \leq K$. Now, we show that it is also true for $n = n' + 1$ and $k = k' + 1$. In this case, we are deciding the number of channels to be assigned to the $(k' + 1)$-th client out of these $n' + 1$ channels. We can assign 0 to $\min\{n_{k'+1}, n' + 1\}$ channels to the $(k' + 1)$-th client. If we assign $x_{k'+1}$ channels to the $(k' + 1)$-th client, the optimal value for equation (4.1) is the sum of $(d_j^+(x_{k'+1}), d_j^-(x_{k'+1}))^T$ and $\vec{f}(n' + 1 - x_{k'+1}, k')$, which is the optimal value for assigning $(n+1-x_{k'+1})$ channels to the first k' clients. Therefore, $\vec{f}(n'+1,k'+1)$ is obtained by considering all possible values of $(d_j^+(x_{k'+1}), d_j^-(x_{k'+1}))^T + \vec{f}(n'+1-x_{k'+1},k')$ for $0 \leq x_{k'+1} \leq \min\{n_{k'+1}, n'+1\}$. Hence, $\vec{f}(n'+1,k'+1)$ is obtained by choosing the lexicographic minimum of$(d_j^+(x_{k'+1}), d_j^-(x_{k'+1}))^T + \vec{f}(n'+1-x_{k'+1},k')$ for $0 \leq x_{k'+1} \leq \min\{n_{k'+1}, n'+1\}$, which is the recursive expression described in the inductive assignment algorithm in Section 4.1.2. Hence, the expression for $\vec{f}(n,k)$ is also true for $n = n'+1$ and $k = k'+1$.

By mathematical induction, this expression is true for all n and k. Therefore, by using the recursive relations for all $\vec{f}(n,k)$, the optimal solution solution can be found from $\vec{f}(N',K)$.

Appendix C

Assignment Problem

The assignment problem is about finding an optimal way to assign N jobs to N workers. The cost of assigning job i to worker j is $c_{i,j}$. Let $x_{i,j}$ be the binary decision variable such that $x_{i,j} = 1$ if job i is assigned to worker j. Otherwise, $x_{i,j} = 0$. The objective is to minimize the total cost of this job assignment. Many problems in engineering and resource management can be formulated as an assignment problem. Therefore, it is a well-known problem in operations research.

The problem can be modelled as follows:

$$\text{Minimize} \quad \sum_{i=1}^{N} \sum_{j=1}^{N} c_{i,j} x_{i,j} \tag{C.1}$$

subject to

$$\sum_{i=1}^{N} x_{i,j} = 1, \quad \forall j \tag{C.2}$$

$$\sum_{j=1}^{N} x_{i,j} = 1, \quad \forall i \tag{C.3}$$

$$x_{i,j} \in \{0, 1\}, \quad \forall i, j \tag{C.4}$$

By the integer solutions property [10], constraint (C.4) can be converted to

$$x_{i,j} \geq 0, \quad \forall i, j. \tag{C.5}$$

Now, the problem is a standard linear programming problem which can be solved with common mathematical software.

Another way to solve the problem is by using the *Hungarian Method* [16]. This algorithm is outlined as below.

To begin with, we let $\mathbf{C}$ be the matrix

$$\mathbf{C} = \begin{pmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,N} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ c_{N,1} & c_{N,2} & \cdots & c_{N,N} \end{pmatrix}. \tag{C.6}$$

Then, we follow the steps below:

1. Subtract the entries of each row of $\mathbf{C}$ by the row minimum. As a result, each row has at least one zero and all entries are nonnegative.

2. Subtract the entries of each column by the column minimum. Now, each row and each column has at least one zero.

3. Select rows and columns across which you draw lines, in such a way that all the zeros are covered and that no more lines have been drawn than necessary.

4. If the number of the lines is N, choose a combination from the modified cost matrix in such a way that the sum is zero. Otherwise, that is, the number of lines is less than N, go to step 5.

5. Find the smallest element which is not covered by any of the lines. Then subtract it from each entry which is not covered by the lines and add it to each entry which is covered by a vertical and a horizontal line. Go back to step 3.

Bibliography

[1] P. R. Barbosa, Z. Wu, and C. R. Nassar. High-performance mimo-ofdm via carrier interferometry. In *IEEE Global Telecommunications Conference, Globecom 2003, Vol: 2*, pages 711–715, Dec. 2003.

[2] S. Borst and P. Whiting. Dynamic rate control algorithms for hdr throughput optimization. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, Volume: 2*, pages 976–985, Apr. 2001.

[3] J. R. Boucher. *Voice Teletraffic Systems Engineering*. Artech House, 1988.

[4] M. Carsoni, G. Immovilli, and M. L. Merani. Admission control in t/cdma systems supporting voice and data applications. *Wireless Communications, IEEE Transactions on, Vol: 1 , Issue: 3*, pages 540–548, July 2002.

[5] A. Charnes and W. W. Cooper. Goal programming and multiple objective optimizations. *European Journal of Operational Research 1*, 1975.

[6] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*, volume 2nd edition. MIT Press, 2001.

[7] M.-S. Do, Y. Park, and J.-Y. Lee. Channel assignment with qos guarantees for a multiclass multicode cdma system. *Vehicular Technology, IEEE Transactions on , Vol: 51 , Issue: 5*, pages 935–948, Sept. 2002.

[8] I. Ghauri and D. T. M. Slock. Linear receivers for the ds-cdma downlink exploiting orthogonality of spreading sequences.

[9] B. Hayes. The easiest hard problem. *American Scientist, Vol. 90*, pages 113–117, Mar-Apr 2002.

[10] F. S. Hillerm and G. J. Lieberman. *Introduction to Operations Research.* McGraw-Hill, 1995.

[11] J. E. Hopcroft, R. Motwani, and J. D. Ullman. *Introduction to Automata Theory, Languages and Computation*, volume 2nd Edition. Addison Wesley, 2000.

[12] J. P. Ignizio. *Introduction to Linear Goal Programming.* SAGE Publications, Inc, 1985.

[13] A. Jalali and P. Mermelstein. Power control and diversity for the downlink of cdma systems. In *2nd International Conference on Universal Personal Communications, 1993, Vol: 2*, Oct.

[14] Y. Jang and S. Cheng. A new dynamic channel assignment strategy in cellular mobile communication systems.

[15] E. Kreyszig. *Advanced Engineering Mathematics.* John Wiley & Sons, Inc.

[16] H. W. Kuhn. The hungarian method for the assignment problems. *Naval Research Logistics Quarterly*, pages 83–97, 1955.

[17] S. J. Leon. *Linear Algebra with Applications*, volume 6th Edition. Prentice Hall, Inc, 2002.

[18] A. Leon-Garcia and I. Widjaja. *Communication Networks Fundamental Concepts and Key Architectures.* McGraw-Hill, 2000.

[19] G. Li. Throughput maximization with buffer constraints in broadband ofdma networks. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.*

(ICASSP 03). 2003 IEEE International Conference on, Vol. 4, pages 197–200, Apr. 2003.

[20] T. M. Lok. Downlink code assignment for throughput or latency optimization in multicarrier cdma systems. In *Personal, Indoor and Mobile Radio Communications, 2003. PIMRC 2003. 14th IEEE Proceedings on, Volume: 2*, pages 1365–1369, Sept. 2003.

[21] T. M. Lok and T. F. Wong. Transmitter and receiver optimization in multicarrier cdma systems. *IEEE Trans. Commun., vol. 48, no. 7*, July 2000.

[22] B. Natarajan, , Z. Wu, C. R. Nassar, and S. Shattil. Large set of ci spreading codes for high-capacity mc-cdma. *IEEE Transactions on Communications, Vol. 52, Issue: 11*, pages 1862–1866, Nov. 2004.

[23] B. Natarajan, C. R. Nassar, and S. Shattil. High-throughput high-performance tdma through pseudo-orthogonal carrier interferometry. *IEEE Transactions on Wireless Communications, Vol. 3, Issue: 3*, pages 689–694, May 2004.

[24] C. Y. Ng and T. M. Lok. A goal programming approach for downlink channel assignment. In *VTC 2004 fall*, Sept. 2004.

[25] C. Y. Ng, H. M. Tse, and T. M. Lok. A goal programming model and schemes for channel assignment in general downlink transmission system. In *IEEE International Conference on Communications, ICC 2005*, May 2005.

[26] R. L. Peterson, R. E. Ziemer, and D. E. Borth. *Introduction to Spread Spectrum Communications*. Prentice Hall, Inc, 1995.

[27] J. G. Proakis and M. Salehi. *Communication Systems Engineering*, volume 2nd Edition. Prentice-Hall, 2002.

[28] X. Qiu, K. Chawla, J. C.-I. Chuang, and N. Sollenberger. Network-assisted resource management for wireless data networks. *Selected Areas in Communications, IEEE Journal on, Volume: 19 Issue: 7*, pages 1222–1234, July 2001.

[29] T. S. Rappaport. *Wireless Communications Principles and Practice*, volume 2nd Edition. Prentice Hall, 2002.

[30] F. Rashid-Farrokhi, K. J. R. Liu, and L. Tassiulas. Downlink power control and base station assignment. *Communications Letter, IEEE, Vol. 1, Issue: 4*, pages 96–98, July 1997.

[31] S. M. Ross. *Introduction to Probability Models*, volume 8th Edition. Academic Press, 2003.

[32] H. M. Saber and A. Ravindran. Nonlinear goal programming theory and practice: A survey. *Computers and Operations Research, Vol. 20*, pages 275–291, 1993.

[33] C. E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J., Vol. 27*, pages 623–656, Oct. 1948.

[34] C. W. Sung and W. S. Wong. Sequential packing algorithm for channel assignment under cochannel and adjacent-channel interference constraint.

[35] H. A. Taha. *Operations Research An Introduction*, volume 6th Edition. Prentice-Hall, 1997.

[36] X. Tian and C. Ji. Qos provisioning with distributed call admission control in wireless networks. In *Communications, 1998. ICC 98. Conference Record.1998 IEEE International Conference on , Volume: 2*, pages 797–801, June 1998.

[37] S. Verdù. *Multiuser Detection*. Cambridge University Press, 1998.

[38] A. J. Viterbi. *CDMA Principles of Spread Spectrum Communication*. Addison Wesley, 1995.

[39] V. Vitsas and A. C. Boucouvalas. Optimization of irda irlap link access protocol. *Wireless Communications, IEEE Transactions on, Volume: 2 Issue: 5,*, Sept. 2003.

[40] M. A. Weiss. *Data Structures and Algorithm Analysis in C*, volume 2th Edition. Addison Wesley, 1999.

[41] D. A. Wiegandt and C. R. Nassar. Higher-speed, higher-performance 802.11a wireless lan via carrier-interferometry orthogonal frequency division multiplexing. In *IEEE International Conference on Communications, ICC 2002, Vol. 1*, 2002.

[42] R. E. Ziemar and R. L. Peterson. *Introduction to Digital Communication.* Prentice Hall, Inc, 2001.