國 立 交 通 大 學

電信工程研究所

博 士 論 文

正交分頻多工無線網路之資源管理

On Resource Management in OFDMA based Wireless Networks

研 究 生：林 淵 斌

指導教授：蘇 育 德 博士

中 華 民 國 九十八 年 七 月

正交分頻多工無線網路之資源管理

Resource Management in OFDMA based Wireless Networks

研　究　生：林淵斌　　　Student: Yuan-Bin Lin
指導教授：蘇育德　博士　　　Advisor: Dr. Yu T. Su

國立交通大學

電信工程研究所

博士論文

A Dissertation
Submitted to Institute of Communications Engineering
College of Electrical and Computer Engineering
National Chiao Tung University
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
in
Communications Engineering
Hsinchu, Taiwan

2009 年 7 月

正交分頻多工無線網路之資源管理

研究生:林淵斌　　　　　　　　　　　指導教授:蘇育德 博士

國立交通大學電信工程研究所

中文摘要

本文旨在探討正交分頻多工無線網路之資源管理並針對各式頻譜與功率分配問題提出最佳與次佳的演算法。我們的演算法採取的是使用者去除的概念來解耦(decouple)多用戶多載波分配的關連性。無線資源分配牽涉到眾多用戶需求條件與系統參數與設計之選擇及實務考量，其複雜度相當高，在大部分情況下無法有最佳的解決方案。本文所考量的情境(scenario)為單一基地台與多個移動台用戶的細胞式通訊系統。

我們主要解決的問題有下列幾項。第一個問題為「在滿足不同用戶之不同傳輸率的要求下使用最少的總功率或能量來分配既有的無線電資源(傳輸功率、能量及次載波)。」其中，我們考慮各通道之增益雜訊比(channel-gain-to-noise-ratio)之不同，提出最佳與次佳的演算法。第二問題則是試圖在單一用戶尖峰傳輸功率的限制下對總加權傳輸率極大化。我們提出兩種次佳的頻譜分配演算法，其中之一利用了對偶分解(dual-decomposition)的方法。第三問題考量了用戶公平性的問題，因此針對用戶傳輸率總乘積之極大化提出一個次佳演算法。傳輸速率總乘積之極大化可使系統在傳輸速率總和增加的同時儘可能地維持各用戶一定的傳輸率！最後一個問題，我們針對多輸入多輸出的通訊系統提出低複雜度的資源管理演算法，使系統在滿足用戶之不同傳輸率要求下能讓總傳輸功率極小化。在此多輸入多輸出的通訊系統中我們採取主對角線塊狀化(block diagonalization)來使其每一載波通道皆可允許多個用戶在無彼此干擾下傳輸。我們開發的演算法亦應用了對偶分解法來解決用戶的移除與選擇。與第二個問題的不同是：每一載波可以保留給多個用戶。基於主對角線塊狀化的特性我們所提出的演算法在保留使用效率高的用戶之同時亦兼顧了保持其空間通道正交的優勢。對於上述各類資源管理問題所提出的最佳或次佳分配解我們均分析了其複雜度並以電腦模擬證明所提出的演算法皆能有甚佳的效能表現。

On Resource Management in OFDMA Based Wireless Networks

Student: Yuan-Bin Lin Advisor: Dr. Yu T. Su

Department of Communications Engineering

National Chiao Tung University

Absract

Algorithms for finding suboptimal and optimal solutions to total power minimization or capacity maximization resource allocation problems in OFDMA-based networks haven been studied by many authors. But the complexities of finding the optimal solution and some suboptimal are prohibitively high and only few numerical examples for low dimension cases can be found. On the other hand, low-complexity suboptimal solutions often give unsatisfactory performance.

In this thesis, we propose optimal and suboptimal resource allocation solutions for OFDMA and MIMO-OFDMA wireless networks. Various design criteria and system constraints are considered. The corresponding complexities for all suboptimal solutions are relatively low while that for the optimal algorithm is only moderate high. We first investigate the problem of transmit power minimization in an OFDMA downlink network subject to user rates and BER requirements constraints. We provide near-optimal and optimal solutions based on the dynamic programming and branch& bound methodologies. The second scenario we consider is a weighted sum rate maximization problem which is solved via dynamic programming and dual decomposition. We then proceed to consider the product rate maximization scenario and present a suboptimal solution. Finally, we consider a total power minimization problem for a MIMO-OFDMA wireless networks and present a low-complexity solution.

The main concept in our proposed algorithms can be easily applied to obtain a near-optimal solution for many similar multi-constraints optimization problem with low complexity.

Acknowledgement

I would like to express my sincere appreciation to my dissertation advisor, Dr. Yu T. Su, for his enthusiastic guidance, for the latitude he gave me and for his encouragement. Without his suggestions, support, and kindly encouragement, the course of this work cannot be under way successfully. Thanks to the members in *Lab 811*, especially for Tai-Hsiang Chiu and all of my friends for their kindly help. Finally, I would like to thank my family, my parents, my wife, for their encouragement and endless love.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

As the demand for high data rate multi-media wireless communications increases, it also becomes more and more important that one takes into account the energy/spectral efficiency factor in designing an anti-fading transmission scheme for mobile terminals. A fast and proper adaptive algorithm in allocating both the physical and MAC layer resources is essential to provide high quality high rate multiuser transmissions. Because of its robustness against frequency-selective fading and its flexibility in appropriating the transmission resources, the OFDM-based Frequency Division Multiple Access (OFDMA) scheme in which each user is allocated a collection of time slots and sub-carriers for transmission, has been adopted in several industrial wireless communication standards. If the allocation is predetermined and static, there may be unused sub-carriers and time slots if the designated users do not need so many signal dimensions.

When there are limited power and multiple orthogonal channels available for transmitting multiuser/multimedia signal, a proper channel and power allocation scheme is needed to minimize the average power consumption, co-channel interferences while meeting various users and media's rate requirement and maintaining the link quality. For an OFDMA system, this problem is complicated by the fact that a subcarrier (channel)[1] is bad, in deep fade and with low channel signal-to-noise ratio (SNR) for one user may be good (with high channel SNR) for another user. In [1], the authors proposed a sub-

[1]We shall use the terms *subcarrier* and *channel* interchangeably throughout this paper.

optimal multiuser subcarrier/bit allocation scheme which minimizes the total transmit power with rate constraints. They relaxed the discrete-(integer-)rate constraint by allowing time-sharing use of a subcarrier by multiple users–an idealized assumption that was subsequently used by many investigators. [4] considered a continuous-rate version of the same problem but forbad the multiple-user-per-subcarrier scenario and suggested a method for computing the optimal solution. Obtaining the exact optimal solutions to either problem requires high computing complexity and become impractical for large channel and/or user constraints. There are many works that studied variations and extensions of [1] or [4]; each deals with different objective function (maximizing weighted sum rate [9], utility [12]), constraint (fairness [7], proportional rates [8]), or scenario (e.g., multi-cell [10], relay-aided [7]). A survey on various dynamic resource allocation (RA) solutions was recently reported in [6].

In Chapter 3 we present two efficient algorithms for solving the problems of [1] and [4], i.e., efficient subcarrier, power and rate assignment schemes that satisfy multi-user multi-media requirements with the minimum total transmitted power are given. The first algorithm uses a dynamic programming (DP) approach; it is simple and offers near-optimal performance. The second algorithm invokes the branch-and-bound (B&B) principle, uses a good initial bound and tight lower bounds along with some complexity-reduction techniques. It gives the optimal solution with a moderate increase of complexity. Our discourse concentrates on the continuous-rate case but both algorithms can be used for the discrete-rate case with a minor modification (see Section IV.D). It is not difficult to see that, through suitable modifications, our algorithms can also be applied to solve a similar RA problem of maximizing the aggregated throughput or weighted sum rate with individual power constraints. The rest of this paper is organized as follows. The ensuing section describes the operation scenarios of concern and gives an optimization problem formulation. Section III presents the proposed DP-based resource allocation algorithm and the B&B-based approach is given in Section IV. We also derive some useful properties and suggest design guidelines there. Numerical performance of the proposed algorithm and some existing suboptimal algorithms is presented in Section V. Finally, we give concluding remarks in Section VI and derive an optimal mono-rate (single user) power allocation (OMPA) algorithm in Appendix A.

Most of the previous approaches show how to efficiently maximize total transmission or minimize the total transmitted power under the related constraints of system and users. In addition to the issue that to minimize total transmitted power under all users' constraints can ensure the benefits of users far away from the base station, most formulated problems and the corresponding solutions are focus on the efficiency issue such that the users closer to the base station or with higher power capability will get the most resource/benefits. The fairness issue in resource allocation is addressed in Chapter 4. First, we discuss the problem of maximizing the weighted sum rate under uplink users' power constraints. A similar weighted downlink sum rate maximization problem with a total power constraint has be investigated and the corresponding optimal solution was known [4]. The problem of maximizing the ergodic rates was discussed in [14].

We propose two efficient suboptimal resource allocation algorithms for the weighted sum rate maximization problem. The first algorithm is a modification of the DPRA algorithm, replacing the original cost function and the OMPA algorithm. The second solution exploits the dual decomposition method in convex optimization theory. The resulting algorithm requires lower complexity but suffers from minor performance degradation. In the second part of chapter 4, the proportion fairness is considered. The distributed users can negotiate via the BS to make their decisions on the subcarrier usage cooperatively such that all users jointly agreements are made. This kind of cooperative problem motivates us to apply the game theory and especially cooperative game theory can achieve the fairness and maximize the overall system rate [15], [16], and [17].

Fair-rate allocation for classical OFDMA systems based on the Nash bargaining solution (NBS) have been recently considered in [16]. As the proposed solution was too computational intensive, the number of users was limited to 8 or less. Our DP-based algorithm is much simpler while its performance is super. The corresponding optimal solution can still be obtained by the BBRA approach with minor modifications.

Finally, we extend our investigation to MIMO-OFDMA systems, focusing on the total transmitted power minimization under users' BER and rate requirements. The multiple antennas at the base station are built for spatial multiplexing of transmissions to multiple users in the same time subcarrier. Our precoding scheme is based on the block diagonalization method. With block diagonalization, each user's precoding matrix is designed such that the transmitted signal of that user lies in the null space of all other remaining users' and multiuser interference is pre-eliminated. The resource allocation problem in a MIMO OFDMA system using block diagonalization becomes that of selecting users who can share the same subcarrier for all subcarriers. [19] has discussed the similar power minimization problem without user selection over each single carrier. [21] proposed a user selection scheme based on the users' channel conditions and correlations for MISO systems. Both considered only the single carrier case whence frequency assignment is not needed. The propose of Chapter 5 is to present a dual decomposition based low-complexity suboptimal solution which employs a correlation-based user selection scheme to simultaneously complete the task of user selection and subcarrier assignment in a MIMO OFDMA system.

Chapter 2

Review of Some Optimization Methodologies

2.1 Introduction to Dynamic Programming

The *dynamic programming (DP)* was coined by Bellman [22] to describe the techniques which he brought together to study a class optimization problems involving sequences of decisions. There have been many applications and further developments since its inception. In this thesis, we focus on the situations where decisions are made in stages.

2.1.1 The Basic Problem

We now formulate a general multi-stage statistical decision problem under stochastic uncertainty. This problem, which is called *basic*, is very general. In particular, it is not necessary to require that the state, control, or random parameter take a finite number of values or belong to a space of n-dimensional vectors. An attractive aspect of dynamic programming is that its applicability depends very little on the nature of the state, control, and random parameter spaces. For this reason, it is convenient to proceed without any assumptions on the structure of these spaces.

We are given a discrete-time dynamic system

$$x_{k+1} = f_k(x_k, u_k, w_k), \quad k = 0, 1, ..., N-1 \tag{2.1}$$

where

k	the index of discrete time,
x_k	the state of the system and summarized past information that is relevant for future optimization, $x_k \in S_k$,
u_k	the control or decision variable to be selected at time k, $u_k \in C_k$,
w_k	a random parameter,
N	the horizon or number of times control which is applied,
f_k	a function that describes the system and in particular the mechanism by which the state is updated.

The control variable u_k is constrained to take values in a given nonempty subset $U(x_k) \subset C_k$, which depends on the current state x_k; that is, $u_k \in U_k(x_K)$ for all $x_k \in S_k$ and k. We consider the class of policies (also called control laws) that consist of a sequence of functions

$$\pi = \{\mu_0, ..., \mu_{N-1}\} \tag{2.2}$$

where μ_k maps state x_k into controls $\mu_k = \mu_k(x_k)$ and is such that $\mu_k(x_k) \in U_k(x_k)$ for all $x_k \in S_k$. Such policies are called *admissible*.

Given an initial state x_0 and an admissible policy $\pi = \{\mu_0, ..., \mu_{N-1}\}$, the states x_k and disturbances w_k are random variables with distributions defined through the system equation

$$x_{k+1} = f_k(x_k, \mu_k(x_k), w_k), \quad k = 0, 1, ..., N-1 \tag{2.3}$$

Thus, for given functions, g_k, $k = 0, 1, ..., N$, the expected cost of π starting at x_0 is

$$J_\pi(x_0) = E\left\{ g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, \mu_k(x_k), w_k) \right\} \tag{2.4}$$

where the expectation is taken over the random variables w_k and x_k. An optimal policy π^* is one that minimizes the cost; that is,

$$J_{\pi^*}(x_0) = \min_{\pi \in \Pi} J_\pi(x_0) \tag{2.5}$$

where Π is the set of all admissible policies.

2.1.2 The Dynamic Programming Algorithm

The dynamic programming technique is built upon a very simple idea, the *principle of optimality* which can be formally stated by

Principle of Optimality

Let $\pi^* = \{\mu_0^*, \mu_1^*, ..., \mu_{N-1}^*\}$ be an optimal policy for the basic problem, and assume that when using π^*, a given state x_i occurs at time i with positive probability. Consider the subproblem whereby we are at x_i at time i and wish to minimize the "cost-to-go" from time i to time N

$$E\left\{g_N(x_N) + \sum_{k=i}^{N-1} g_k(x_k, \mu_k(x_k), w_k)\right\} \tag{2.6}$$

Then the truncated policy $\{\mu_i^*, \mu_{i+1}^*, ..., \mu_{N-1}^*\}$ is optimal for this subproblem. ■

The principle of optimality suggests that an optimal policy can be constructed in piecemeal fashion, first constructing an optimal policy for the "tail problem" involving the last stage, then extending the optimal policy to the "tail problem" involving the last two stages, and continuing in this manner until an optimal policy for entire problem is constructed. The dynamic programming algorithm is based on this idea: it proceeds sequentially, by solving all the tail subproblems of a given (time) length.

We now state the dynamic programming algorithm for the basic problem.

The Dynamic Programming Algorithm

For every initial state x_0, the optimal cost $J_\pi^*(x_0)$ of the basic problem is equal to $J_0(x_0)$, given by the first step of the following algorithm, which proceeds forward in time from stage 1 to stage N:

$$J_0(x_0) = g_0(x_0),$$

$$J_k(x_k) = \min_{u_k \in U_k(x_k),\ w_k} E\{g_k(x_k, u_k, w_k) + J_{k-1}(f_x(x_k, u_k, w_k))\},\ k = 1, ..., N \tag{2.7}$$

where the expectation is taken with respect to w_k, which depends on x_k and u_k. Furthermore, if $u_k^* = \mu_k^*(x_k)$ minimizes the right side of (2.7) for each x_k and k, the policy $\pi^* = \{\mu_0^*, \mu_1^*, \cdots, \mu_{N-1}^*\}$ is optimal. ■

The term "dynamic programming" coined by Richard Bellman was originally referred to the process of solving problems where one needs to find the best decisions one after another. It was later refined to referring to the general approach of nesting smaller decision problems inside larger decisions. Equivalently, DP means simplifying a complicated problem by breaking it down into simpler subproblems in a recursive manner. Note that a dynamic programming algorithm is capable of obtaining the optimal solution only if the "cost-to-go" can be decomposed into the recursive form of (2.6).

2.2 The Branch and Bound Principle

Solving an NP-hard discrete optimization problem is often an immense job requiring a very efficient algorithm, and the Branch and Bound (B&B) paradigm is one of the main tools used in constructing such a solution. A B&B method searches for the best solution in the complete space of solutions according to a given problem. However, explicit enumeration is normally impossible due to the exponentially increasing number of potential solutions. The use of bounds for the function to be optimized combined with the value of the current best solution enables the algorithm to search parts of the solution space only implicitly.

At any point during the solution process, the status of the solution with respect to the search of the solution space is described by a pool of yet unexplored subset of this and the best solution found so far. Initially only one subset exists, namely the complete solution space, and the best solution found so far is ∞. The unexplored subspaces are represented as nodes in a dynamically generated search tree, which initially only contains the root, and each iteration of a classical B&B algorithm processes one such node. The iteration has three main components: selection of the node to process, bound calculation, and branching. In Fig. 2.1, the initial situation and the first step of the process are illustrated.

The sequence of these may vary according to the strategy chosen for selecting the next node to process. If the selection of next subproblem is based on the bound value of the subproblems, then the first operation of an iteration after choosing the node is branching. For each of these, it is checked whether the subspace consists of a single solution, in which case it is compared to the current best solution keeping the best of these. Otherwise the bounding function for the subspace is calculated and compared to the current best solution. If the subspace cannot contain the optimal solution, the whole subspace is discarded. The search terminates when there are no unexplored parts of the solution space left, and the optimal solution is then the one recorded as "current best"

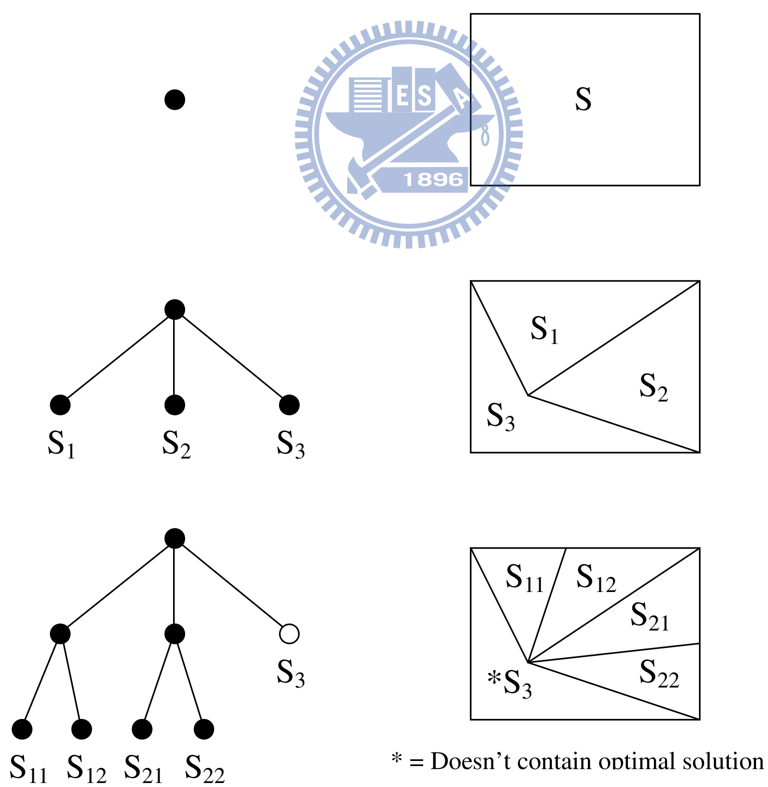

Figure 2.1: Illustration of the search space for a B&B procedure.

2.2.1 Terminology and General description

In the following subsection, we consider minimization problems - the case of maximization problems can be dealt with similarly. The problem is to minimize a function $f(x)$ of variables $(x_1 \ldots x_n)$ over a region of *feasible solutions*, S:

$$\min_{x \in S} f(x)$$

The function f is called the *objective function* and may be of any type. The set of feasible solutions is usually determined by general conditions on the variables, e.g. that these must be non-negative integers or binary, and special constraints determining the structure of the feasible set. In many cases, a set of *potential solutions*, G, containing S, for which f is still well defined. A function $g(x)$ often defined on G (or S) with the property that $g(x) \leq f(x)$ for all x in S arises naturally. Both S and G are very useful in the B&B context. Fig. 2.2 illustrates the situation where S and G are intervals of real numbers.

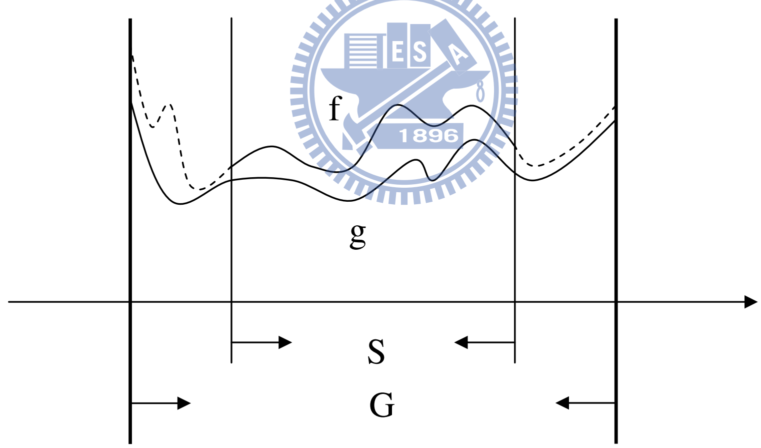

Figure 2.2: The relation between the bounding function g and the objective function f on the sets S and G of feasible and potential solutions of a problem.

2.2.2 Bounding Function

The bounding function is the key component of any B&B algorithm in the sense that a low quality bounding function cannot be compensated for through good choices of branching and selection strategies. Ideally the value of a bounding function for a given subproblem should equal the value of the best feasible solution to the problem, but on account of obtaining this value is usually in itself NP-hard, the goal is to come as close as possible using only a limited amount of computational effort. A bounding function is called *strong*, if it in general gives values close to the optimal value for the subproblem bounded, and *weak* if the values produced are far from the optimum. One often experiences a trade off between quality and time when dealing with bounding functions: The more time spent on calculating the bound, the better the bound value usually is. It is normally considered beneficial to use as strong a bounding function as possible in order to keep the size of the search tree as small as possible.

Bounding functions naturally arise in connection with the set of potential solutions G and the function g mentioned in above. Due to the fact that $S \subseteq G$, and that $g(x) \leq f(x)$ on G, the following is easily seen to hold:

$$\min_{x \in G} g(x) \leq \left\{ \begin{array}{l} \min_{x \in G} \quad f(x) \\ \min_{x \in S} \quad g(x) \end{array} \right\} \leq \min_{x \in S} f(x) \tag{2.8}$$

If both of G and g exist there are now choices between three optimization problems, for each of which the optimal solution will provide a lower bound for the given objective function. The "skill" here is of course to chose G and/or g so that one of these is easy to solve and provides tight bounds.

2.2.3 Branching Rule

All branching rules in the context of B&B can be seen as subdivision of a part of the search space through the addition of constraints, often in the form of assigning values to variables. Convergence of B&B is ensured if the size of each generated subproblem is smaller than the original problem, and the number of feasible solutions to the original problem is finite. Normally, the subproblems generated are disjoint - in this way the problem of the same feasible solution appearing in different subspaces of the search tree is avoided.

2.2.4 Strategies for Selecting Next Subproblem

The strategy for selecting the next live subproblem to investigate usually reflects a trade off between keeping the number of explored nodes in the search tree low, and staying within the memory capacity of the computer used.

If one always selects among the live subproblems one of those with the lowest bound, called the *best first search* strategy, *BeFS*. Fig. 2.3 shows a small search tree -the numbers in each node corresponds to the sequence. A subproblem P is called critical if the given bounding function when applied to P results in a value strictly less than the optimal solution of the problem in question. Nodes in the search tree corresponding to critical subproblems have to be partitioned by the B&B algorithm no matter when the optimal solution is identified - they can never be discarded by means of the bounding function. Since the lower bound of any subspace containing an optimal solution must be less than or equal to the optimum value, only nodes of the search tree with lower bound less than or equal to this will be explored.

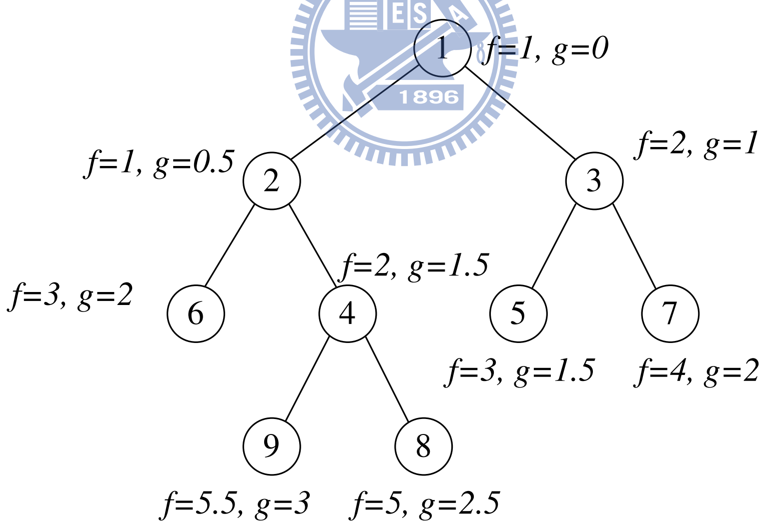

Figure 2.3: Search strategies in B&B: the Best-First Search.

Even though the choice of the subproblem with the current lowest lower bound makes good sense also regarding the possibility of producing a good feasible solution, memory problems arise if the number of critical subproblems of a given problem becomes too large. The situation more or less corresponds to a *breath first search* strategy, *BFS*, in which all nodes at one level of the search tree are processed before any node at a higher level. Fig. 2.4 shows the search tree with the numbers in each node corresponding to the BFS processing sequence. The number of nodes at each level of the search tree grows exponentially with the level making it infeasible to do breadth first search for larger problems.

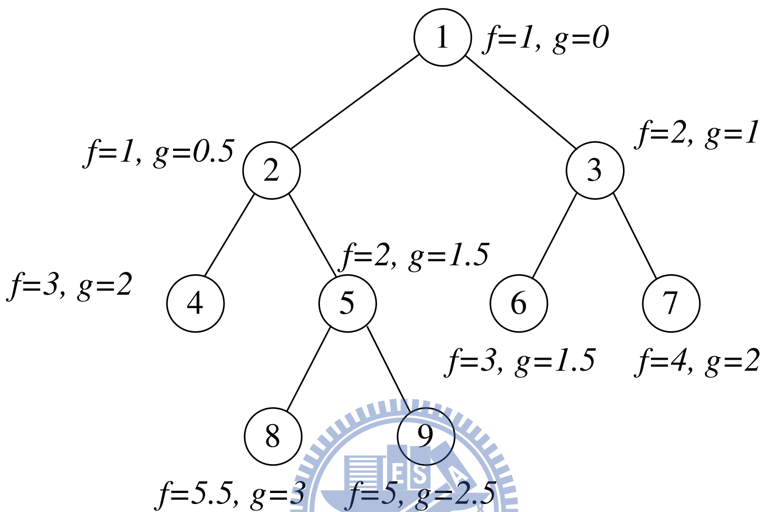

Figure 2.4: Search strategies in B&B: the Breath First Search.

The alternative used is a *depth first search* strategy, *DFS*. Here a live node with largest level in the search tree is chosen for exploration. Fig. 2.5 shows the DFS processing sequence number of the nodes. The memory requirement in terms of number of subproblems to store at the same time is now bounded above by the number of levels in the search tree multiplied by the maximum number of children of any node, which is usually a quite manageable number. An advantage from the programming point of view is the use of recursion to search the tree - this enables one to store the information about the current subproblem in an incremental way, so only the constraints added in connection with the creation of each subproblem need to be stored. The drawback is that if the incumbent is far from the optimal solution, large amounts of unnecessary bounding computations may take place. In order to avoid this, DFS is often combined with a selection strategy which is that exploring the node with the small lower bound first hopefully leads to a good feasible solution.

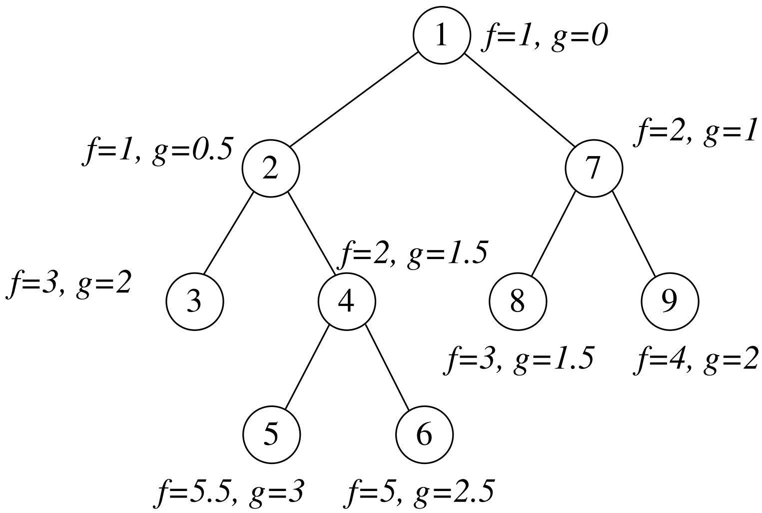

Figure 2.5: Search strategies in B&B: the Depth First Search.

2.3 Dual Decomposition method for Non-convex Optimization

We consider a non-convex optimization problem defined over domain D, the set of all non-negative real $x_j(i)$ for $j = 1, \cdots, K$ and $i = 1, \cdots, N$ such that for each i at most one $x_j(i)$ is positive for $k = 1, \cdots, K$, as follows

$$\begin{aligned} \max_{\{\mathbf{x_j}\}} \quad & \sum_{i=1}^{N}\sum_{j=1}^{K} f_{ij}(x_j(i)) = \sum_{i=1}^{N}\sum_{j=1}^{K} \log_2(1 + x_j(i)a_{j,i}) \\ \text{s.t.} \quad & \sum_{i=1}^{N} x_j(i) \leq P_j \\ \text{where} \quad & \mathbf{x_j} = \{x_j(1)x_j(2)\cdots x_j(N)\} \end{aligned} \tag{2.9}$$

Then the Lagrangian of the above problem can be represented as

$$\mathcal{L}\{x_j(i), \Lambda\} = \sum_{j=1}^{K}\sum_{i=1}^{N} f_{ij}(x_j(i)) - \sum_{j=1}^{K} \lambda_j \left(\sum_{i=1}^{N} x_j(i) - P_j \right) \tag{2.10}$$

where $\Lambda = (\lambda_1, \lambda_2, \cdots, \lambda_K)$. The Lagrange dual function is given by

$$g(\Lambda) = \max_{\{\mathbf{x_j}\}\in\mathbf{D}} \quad \mathcal{L}\{x_j(i), \Lambda\} \tag{2.11}$$

The maximization of $\mathcal{L}$ can be decomposed into N independent optimization problems given by

$$g'_i(\Lambda) = \max_{\{\mathbf{x_j}\}\in\mathbf{D}} \left\{ \sum_{j=1}^{K} f_{ij}(x_j(i)) - \sum_{j=1}^{K} \lambda_j x_j(i), \right\}, \; i = 1, \cdots, N \tag{2.12}$$

The Lagrange dual function can be reformulated as

$$g(\Lambda) = \sum_{i=1}^{N} g'_i(\Lambda) + \sum_{j=1}^{K} \lambda_j P_j. \tag{2.13}$$

With a fixed Λ, the argument on the right hand side of (2.12) becomes a convex function of $\{\mathbf{x_j} = (x_j(1)\ x_j(2)\ \cdots\ x_j(N))\}$. As a result, we can take the derivative of the above function with respect to $x_j(i)$ and obtain the $g'_i(\Lambda)$ maximization solution

$$x_j(i) = \left(\lambda'_j - \frac{1}{a_{j,i}}\right)^+, \tag{2.14}$$

where $\lambda'_j = 1/(\log 2 \cdot \lambda_j)$ and $(t)^+ \stackrel{def}{=} \max(0, t)$.

Since for each $i = 1, 2, \cdots, N$ only one $x_j(i)$ can be positive, we search over all K possible user assignments for $i = 1, 2, \cdots, N$, and decide that $x_{j_o}(i) > 0$, where j_o and $g'_i(\Lambda)$ are

$$\begin{aligned} j_o &= \arg\max_{1\le j\le K} \left[\log_2(1 + x_j(i)a_{j,i}) - \lambda_j \left(\lambda'_j - \frac{1}{a_{j,i}}\right)^+ \right] \\ g'_i(\Lambda) &= \max_{1\le j\le K} \left[\log_2(1 + x_j(i)a_{j,i}) - \lambda_j \left(\lambda'_j - \frac{1}{a_{j,i}}\right)^+ \right] \end{aligned} \tag{2.15}$$

We need to modify $\{\lambda_\mathbf{k}\}$ to meet the constraints $\sum_{i=1}^{N} x_j(i) = P_j$. Even if the constraints are satisfied there is no guarantee that the solution is optimal unless a convergence criterion is in place.

2.4 Efficient Suboptimal Non-convex Optimization via Dual Decomposition

For the dual decomposition method just discussed, the process to obtain the optimal solution consists of the following main steps. Step 1. fixed a multiplier vector $\{\lambda_j\}$. Step 2. for each $1 \le i \le N$, decide the index j^* whose $\{x_{j*}(i)\}$ is maximal among other values $\{x_j(i)\ \ j \ne j^*\}$ which will be forced to zero. In Step 3, based on the result obtained in Step 2 check if this current multiplier vector is optimal, i.e. if the constraints in 2.9 are met. If the answer is negative, the search of multiplier vector is needed.

The main ingredients of the above method are (i) releasing the constraints in (2.9) initially and by applying the dual decomposition approach to obtain a local optimal solution which takes into account the other N constraints such that only one $x_j(i)$ is positive among $1 \le j \le K$ and $i = 1, 2, \cdots, N$, and (ii) finding the multiplier vector used in the dual decomposition approach to meet the constraints (2.9). The relation between the decision of which one $x_j i$ can be positive and a different given multiplier vector is not obvious such that the search of the optimal multiplier is complicated.

Based on these discussion, we propose an efficient suboptimal algorithm via dual decomposition. The main concept in our algorithm is that we release the N constraints in which each constraint denotes only one $x_j(i) > 0,$ for $1 \le j \le K$ to replace releasing K constraints $\{\sum_{i=1}^{N} x_j(i) \le P_j,$ for $1 \le j \le K\}$. In other words, we extend the domain $\mathcal{D}$ such that $x_j(i)$ can be positive for $1 \le j \le K, 1 \le i \le N$.

Then run a finite number of iterations which at most is to be N in order to taking into count some constraint among the previously released constraints. Within each iteration, we can get a multiplier vector under the constraints $\{\sum_{i=1}^{N} x_j(i) \le P_j,$ for $1 \le j \le K\}$ and try to meet one of the released constraints before current iteration. In detail, we exploit this multiplier vector into dual decomposition and get the efficiency value for

$1 \le i \le N$ and $1 \le j \le K$ as following.

$$\epsilon_{j,i} = f_{ij} - \lambda_j(x_j(i)) \tag{2.16}$$

In addition, we have to decide which one among released constraints previously to be meet with each iteration. We take the sum of efficient values over all $1 \le j \le K$ for all released constraints. We select i^* whose sum efficiency value ν_i^* is largest which is given by

$$\begin{aligned} i^* &= \arg\max_{i \in \mathcal{S}} \nu_i^* \\ &= \arg\max_{i \in \mathcal{S}} \sum_{j=1}^{K} \epsilon_{j,i} \end{aligned} \tag{2.17}$$

where $\mathcal{S}$ denotes a set whose elements represent the indices of released constraints. We decide which $x_j(i^*)$ can be positive in the similar way in optimal dual decomposition method by (2.15). The difference between ours and optimal dual decomposition method occurs. In our algorithm, we just take $i = i^*$ into (2.15) not $i = 1, 2, \cdots, N$. In addition, the constraint $i = i^*$ will be not discussed anymore. As a result, after N iterations, the final solution will meet all the constraints. Detailed procedure is given in the following table.

Step 1:	(Extend the domain $\mathcal{D}$
	by release constraints on only one $x_j(i)$ is positive
	among $1 \leq j \leq K$ for all $1 \leq i \leq N$)
	Set $C(j) = \{i \mid 1 \leq i \leq N\}$, for $1 \leq j \leq K$,
	where $m \in C(j)$ represents that $x_j(m)$ can be positive
	set $\mathcal{S} = \{1, 2, \cdots, N\}$ and $t = 0$
Step 2:	(Select one of released constraints for taking into count)
	Based on $C(j)$ and $\sum_{i=1}^{N} x_j(i) \leq P_j$ for all $1 \leq j \leq K$,
	get a multiplier $\lambda_{\mathbf{j}}^{\mathbf{t}}$ for maximization $\sum_{i=1}^{N} f_{ij}(x_j(i))$
	if $t < N$
	$i^* = \arg_{i \in \mathcal{S}} \nu_i$
	Set $S = S \backslash \{i^*\}$, $t = t + 1$, then goto Step 3.
	else goto Step 5.
	end
Step 3:	Decide which $x_j(i)$ can be positive for all $1 \leq j \leq K$,
	when given $i = i^*$, $j^* = \arg\max \epsilon_{j,i^*}$.
	goto Step 4
Step 4:	(Modify set $C(j)$ for $1 \leq j \leq K$.)
	for $j = 1 : K$
	if $j \neq j^*$ **then** $C(j) \backslash \{i^*\}$ **end**
	end
	goto Step 2
Step 5:	(Output) a suboptimal solution is obtained

Table 2.1: An efficient suboptimal optimization method for non-convex optimization via dual Decomposition.

Chapter 3

Power Minimization Resource Allocation Algorithms

This chapter is concerned with a simple but general scenario in which the object is to allocate the available multiple carriers to multiple users such that the total transmit power is minimized while each user's rate and BER requirements are met. As will become clear in later chapters, the methodologies presented in this chapter serve as prototype approaches which can be easily modified to solve problems arose from other similar system design criteria.

3.1 Assumptions and Problem Formulation

3.1.1 Basic Assumptions

We assume that there are N orthogonal subcarriers. $\mathbf{C} = \{1, 2, \cdots, N\}$ and d user data streams with the rate requirements $\mathbf{R} = \{R_j, j = 1, 2, \cdots, d\}$ to be transmitted over an OFDMA downlink, where the required transmission rate of user j is denoted by R_j. We further assume that the base station and d user terminals are each equipped with single antenna. The base station assigns a set of subcarriers to each user and determines the power and number of bits per OFDM symbol to be transmitted on each subcarrier. The cyclic prefix (guard interval) is long enough to remove all intersymbol interference caused by multipath propagation. In addition, sharing the same subcarrier by different users is not allowed. The base station's resource allocation decision is sent to all users through a separate control channel. At each terminal, user can demodulate the signals over those subcarriers assigned to it. Denoting by c_{ij} the bit rate of the ith subcarrier which serves the jth user, we can express the maximum achievable rate (capacity) c_{ij} using transmitted power p_{ij} as

$$c_{ij} = W_i \log_2\left(1 + \frac{p_{ij}|h_{ij}|^2}{\sigma_{ij}^2}\right),\ 1 \le i \le N,\ 1 \le j \le d \tag{3.1}$$

where W_i is the bandwidth for channel i, $|h_{ij}|^2$ and σ_{ij}^2 denote the channel gain and noise power of the ith channel which serves the jth user. The normalized capacity (rate) r_{ij} of the ith channel when used for serving the jth user is given by

$$r_{ij} = \frac{c_{ij}}{W_i} = \log_2\left(1 + \frac{|h_{ij}|^2 p_{ij}}{\sigma_{ij}^2}\right) = \log_2\left(1 + a_{ij}p_{ij}\right), \tag{3.2}$$

where $a_{ij} = |h_{ij}|^2/\sigma_{ij}^2$ is the corresponding channel gain-to-noise ratio (GNR).

3.1.2 Problem Formulation

Given the multi-user transmission requirements and channel state information (i.e., a_{ij}'s), one would like to find the subcarrier assignment and power allocation that minimize the total transmitted power. We define the $N \times d$ subcarrier assignment matrix $\mathbf{A} = [\mathbf{A_{ij}}]$ by $A_{ij} = 1$ if the ith subcarrier is used to transmit the jth user; otherwise, $A_{ij} = 0$. As a subcarrier can only serve one user at a given time interval, A_{ij} is either 1 or 0 and a legitimate channel assignment matrix $\mathbf{A}$ must satisfy

$$\sum_{j=1}^{d} A_{ij} \le 1,\ \ \sum_{i=1}^{N} A_{ij} \ge 1,\ 1 \le i \le N,\ 1 \le j \le d \tag{3.3}$$

For the downlink case, all signals are transmitted from the same base station, hence only the total transmitter power will be considered. Let $\mathbf{P}$ be the power allocation matrix with (i,j)th entry, p_{ij}, then the problem of concern becomes

$$\min_{\mathbf{P},\mathbf{A}} \sum_{i=1}^{N}\sum_{j=1}^{d} A_{ij}p_{ij} \ \ \text{s.t.} \sum_{i\in C(j)} r_{ij} \ge R_j,\ \ \sum_{j=1}^{d} A_{ij} \le 1$$
$$\text{where}\ \ C(j) = \{i | A_{ij} = 1, 1 \le i \le N\} \tag{3.4}$$

Although in reality there is a total power constraint $\sum_{i=1}^{N}\sum_{j=1}^{d} p_{ij} \leq P_c$, we shall not consider this constraint to begin with. Solving the problem with the total power constraint follows a two-step procedure. In the first step we solve the unconstrained problem to obtain the required optimal total power and then check if the solution meets the total power constraint. The problem is solved if the constraint is satisfied; otherwise the problem does not have an admissible solution and one is forced to go to the second step. In the second step, one can prioritize users' transmission requests, modify (decrease) some rate requirements according to the corresponding latency requirements, or settle with a suboptimal channel/power allocation to accommodate the total power constraint. Which of these options is chosen depends on other system design considerations and the final solution is likely to be obtained by an *outer* iterative process. As far as this paper is concerned, however, the total transmit power constraint will not be discussed henceforth.

In the next section, we adopt a DP approach to derive a simple and practical solution which requires much lower complexity than that of [4] and, more importantly, offers near-optimal performance.

3.2 Dynamic Programming based Near-optimal Resource Allocation

When $d = 1$ the optimal solution to (4) can be obtained by a water-filling process (for parallel Gaussian channels). The water-filling level, however, is difficult to determine. We present a very efficient algorithm called OMPA in Appendix A. Hence if the channel assignment is known, one can determine each user's optimal power allocation by using the proposed OMPA algorithm.

For the general case ($d \neq 1$), an obvious optimal solution to (4) is the exhaustive search over all possible channel assignments with the associated power allocation matrices computed by the OMPA algorithm (or water-filling method) to satisfy all users' rate requirement. Although this algorithm is guaranteed to yield the optimal solution, the searching process is prohibitively complicated, especially if the numbers of users and/or subcarriers are large. An improvement is suggested in [4] which first determines the "water-filling" levels and the channels for each user. Overbooking of channels is inevitable as every one wants the best channels. A complicated process is thus needed to resolve such conflicts and recompute the "water-filling" levels iteratively. Although optimal solution can be found, the complexity is still very high and is practical for small N and d only (e.g., the case $N = 8, d = 2$ was given in [4]). Although [1] considered a discrete-rate scenario the authors relaxed the discrete constraint to find a lower-bound solution of (4) iteratively. The quantized version of this solution gives a suboptimal subcarrier allocation, $\{C(j) : 1 \leq j \leq d\}$, where $C(j)$ is the jth user's serving-channel set (SCS) that consists of the indices of the assigned channels. A single-user rate (bit) allocation algorithm is then applied to each $C(j)$. Numerical behavior of this approach was shown but no comparison with the optimal performance was given.

Other earlier suboptimal proposals [5], [6], [8] for solving (3.4) start with some initial subcarrier (channel) allocation and assign remaining available subcarriers sequentially according to some ad hoc criterion. Since a given channel has different GNRs when serving different users, [8] gives a channel to the user with strongest gain, i.e., the ith subcarrier is assigned to the kth user if $k = \arg\max_{1 \leq j \leq d} a_{ij}$. However, the ordering of the subcarriers or the user is arbitrary and it is highly likely that the best channels for two users are the same, say channel k, but the second best channel for the first user is much better than that for the second user. When the first user obtains channel k the second user can only use its second best channel which is much worse than channel k. If instead, the first user is given its second best channel which is not much worse than channel k while the second user is assigned channel k then the overall performance (required total power) will be much improved. On the other hand, [5] makes an initial SCS size $|C(j)|$ decision based on users' average channel GNRs and rate requirements R_j's. The average GNR ignores frequency selectivity and the resulting algorithm is unlikely to find the optimal solution.

In contrast, our approach begins with the fair initial condition that all users are given the opportunity to take every subcarrier. The proposed channel allocation process consists of a series (N-level) of deletion decisions. At each level, a subcarrier is given to an user and is simultaneously removed from the SCSs of all other users, where the SCS for the jth user at the tth level, $C_t^s(j)$, is the set of all subcarriers allocated to serve user j then. Obviously, our fair initial condition implies that $C_0^s(j) = \{1, 2, \cdots, N\}$, $\forall\, j$. We initially eliminate the constraint $C_t^s(i) \cap C_t^s(j) = \emptyset, \forall\, i \neq j, t = 0, 1, \cdots, N$ and, at stage t, impose the constraint that $t \in C_t^s(j)$ for only one j (i.e., the tth channel can only be in one of SCS's) so that the original single-user-per-subcarrier (SUPS) constraint is eventually re-installed and satisfied. Hence, in a sense what we adopt is a constraint relaxation approach.

In such a sequential assignment process the order of subcarriers may be important as once a subcarrier is assigned, no re-assignment is possible. A reasonable ordering is to sort (re-arrange) the N subcarriers in descending order of their maximum GNR, $a_i^* = \max_{1 \leq j \leq d} a_{ij}$ such that with the new channel order, channel 1 has the best GNR, followed by channel 2, channel 3, $\cdots$, etc. Formally, this channel sorting is the permutation μ on the ordered integer set $\{1, 2, \cdots, N\}$ which satisfies the inequality $a^*_{\mu^{-1}(1)} > a^*_{\mu^{-1}(2)} > \cdots > a^*_{\mu^{-1}(N)}$, where μ^{-1} is the inverse mapping of μ.

Our DP-based algorithm can be described by a d-ary tree in which there are d outgoing branches at the root (initial level) to represent possible assignment of the channel 1. Similarly, every node at any given level (height), say the tth level, has d outgoing branches (to d child nodes), each represents a possible channel-assignment (removal) decision and a tentative channel allocation. The channel allocation is tentative because only t channels are assigned and the remaining $N - t$ channels still belong to all SCSs and unassigned. If we associated each level's decision with a cost, then at the kth level, we shall assign channel k to user i and remove it from the SCSs of all other users (branches) if the associated cost is minimized. Such a decision is equivalent to selecting the ith branch emitted from the surviving node at the $(k-1)$th level as the survival branch while all other $d-1$ branches are terminated.

Given the initial fair channel allocation and the ultimate object of minimizing the required power, the cost for a decision at any level should be the minimum required power for the corresponding tentative channel allocation. Hence if we define the SCS collection at the tth level as $\mathbf{C}_t^s = (C_t^s(1), \cdots, C_t^s(d))$, then the corresponding cost function J_t is

$$J_t\left(\mathbf{C}_t^s\right) = \sum_j g(R_j; C_t^s(j)) \tag{3.5}$$

in which each $g(R_j; C_t^s(j))$ is determined by applying the OMPA algorithm to solve the problem

$$\begin{aligned} \text{Given} \;\; C_t^s(j), \;\; \text{find} \;\; g(R_j; C_t^s(j)) = \min & \sum_{i \in C_t^s(j)} p_{ij} \\ \text{s.t.} & \sum_{i \in C_t^s(j)} r_{ij} \geq R_j. \end{aligned} \tag{3.6}$$

$C_t^s(j)$ for each j is modified at each level so that the subcarrier and power assignment process is guaranteed to end at the Nth level. As the minimum required power $g(R_j; C_t^s(j))$ for each j is a decreasing function of the cardinality $|C_t^s(j)|$ of its SCS, the cost J_t is an increasing function of t. At each level, however, we find the removal of the subcarrier from all but one $C_t^s(j)$ that results in minimum cost (total power) increase. As the collection $\{C_t^s(j)\} = \mathbf{C}_t^s$ allows multiple channel assignments, i.e., $C_t^s(j) \cap C_t^s(k) \neq \emptyset$, if $j \neq k$ and $t < N$, it does not satisfy the constraints (3) of a legitimate channel assignment matrix. But as the subcarriers are assigned to users one by one, at the end of the Nth level, $\{C_N^s(j)\} = \mathbf{C}_N^s$ will correspond to a legitimate one. Therefore, the metric defined by (5)-(6) is simply the minimum total transmit power for a given rate-subcarrier assignment with various degrees of relaxation on the SUPS constraint.

Since a path in the tree that visits the jth child node at the kth level implies a channel assignment that gives the kth subcarrier to the jth user, an N-level path would represent a complete channel allocation. But not all paths are legitimate for a path may assign no serving-channel to an user. In particular, if at the end of the tth level there are still more than $N-t$ users without any serving-channel, i.e., whose SCS cardinality is equal to $N-t$, then there will be at least one user with an empty SCS at the end of the Nth level. To avoid such a possibility and rule out all illegitimate channel assignments, we modify the cost function as

$$\begin{aligned} J_t(\mathbf{C}_t^s) &= \min_{1\le k\le d} \left\{ \sum_{j=1}^{d} g(R_j, C_t^s(j;k)\ ) \right. \\ &\quad \left. + \omega_t \left[\sum_{j=1}^{d} \delta(N-t-|C_t^s(j;k)|) \right] \right\} \\ &\stackrel{def}{=} \min_{1\le k\le d} J_t^k(\mathbf{C}_t^s) \end{aligned} \tag{3.7}$$

where

$$C_t^s(j;k) = \begin{cases} C_{t-1}^s(j) &, \quad j=k \\ C_{t-1}^s(j)\setminus\{t\} &, \quad j\neq k \end{cases} \tag{3.8}$$

$$\delta(x) = \begin{cases} 1 &, \quad x=0 \\ 0 &, \quad \text{otherwise} \end{cases}$$

and

$$\omega_t(x) = \begin{cases} 0 &, \quad x \le N-t \\ \infty &, \quad x > N-t \end{cases} \tag{3.9}$$

By adding the weight function $w_t(\cdot)$ in the cost function, we avoid continuously assigning channels to some users while other users might not be able to obtain any channel, although the probability of such an event is almost zero so long as $N>d$ and the GNR distributions $\{a_{ij}, i=1,2,\cdots,N\}$ for each user are independent.

The resulting DP-based resource allocation (DPRA) algorithm, unlike other approaches [1][8][6], accomplishes channel and power (rate) allocations simultaneously and is listed in Table 3.1. Early terminations and computational complexity reduction are possible if certain conditions are satisfied; see *Guidelines 4, 5* in the next section.

Step 1:	(Channel-sorting) Given N, d, a_{ij} and $\mathbf{R}$, find $a_i^* = \max_{1 \le j \le d} a_{ij}$ and re-arrange the channel indexes by decreasing magnitude of the maximum GNR such that $a_1^* > a_2^* > \cdots > a_N^*$ with the new channel indexes.
Step 2:	(Initial channel allocation) Set $C_0^s(j) = \{i \mid 1 \le i \le N\}$, for $1 \le j \le d$.

Step 3:	(Sequential channel-power-rate assignment)
	for $t = 1 : N$
	k^*=arg $\min_{1 \le k \le d} J_t^k(\mathbf{C}_t^s)$
	$J_t(\mathbf{C}_t^s) = J_t^{k^*}(\mathbf{C}_t^s)$
	  for $j = 1 : d$
	    if $j = k^*$ **then** $C_t^s(j) = C_{t-1}^s(j)$
	    else $C_t^s(j) = C_{t-1}^s(j) \backslash \{t\}$
	  end
	end

Step 4:	(Output) The final channel allocation is the Nth level SCS collection $\mathbf{C}_N^s$. The power-rate allocation is obtained while computing $J_N(\mathbf{C}_N^s)$ through (5)-(7).

Table 3.1: A dynamic programming based resource allocation (DPRA) algorithm

3.3 An Optimal Resource Allocation Algorithm

The N-level tree shown in Fig. 1 is a graphic representation of the solution space of (4). The tree contains all possible–legitimate or illegitimate–channel assignments. At each level we allocate a channel so that each "complete" path L from the root node to a leaf node represents a candidate assignment and can be denoted by $L = (b_1, b_2, \cdots, b_N)$, where b_i is the ith node visited by the path and, for brevity, the initial (root) node is not included in the notation. A partial path $l_n = (b_1, b_2, \cdots, b_n), n < N$ is thus defined as the part of a complete path that starts at the root node and ends at some internal node.

Searching over the complete tree can certainly lead to the optimal solution but the complexity is of exponential order. The DPRA algorithm calls for the elimination of $d-1$ child nodes at each level and promises to finish the tree-search process in N stages. As will be shown in Section V, this approach is very efficient in that it yields near-optimal solution with low complexity. However, there is no guarantee that the optimal solution will be obtained as it is possible that the optimal channel assignment path is discarded somewhere along the way, especially if SNR is low. Many other techniques can be used to reduce the prohibitive high complexity of searching the total solution space. We employ a simple linear programming technique called branch-and-bound (B&B) which has the potential of significant complexity reduction if the bounds are properly chosen. Besides presenting novel tight bounds, we also suggest a subcarrier-sorting procedure, which is crucial in reducing the search complexity, use a good initial upper bound and derive some useful properties and guidelines for further complexity reductions.

3.3.1 A Branch-and-bound Approach

In the search tree shown in Fig. 1, each parent node has d child nodes to enclose all possible solutions. Similar to our description of the DPRA algorithm, a path in Fig. 1 that passes through the jth child node at the kth level of the tree (i.e., $b_k = j$) represents a channel assignment that gives the kth subcarrier to the jth user and a (complete) path is legitimate only if it visits every candidate child node at least once. The B&B paradigm needs an upper bound B_u on any $v(l_N)$ (i.e., legitimate channel assignment or complete path) and a lower bound $B_l(l_t)$ associated with each partial path l_t of length t. The use of the upper bound for the cost (minimum required total power) combined with the lower bound which represents the current best solution value (associated with a partial path) enables the algorithm to prune the non-promising subtrees rooted at certain nodes and search parts of the complete tree only. These bounds should be updated as soon as possible to accelerate the searching process but the initial upper bound often plays an important role in the reducing the search complexity. A weak bound will not be capable of eliminating many visits to nodes that lie outside of the correct (optimal) path. To find a tight lower bound, we need the following fundamental definition.

Definition 1. *The node value (cost) $v(l_t)$ of an internal node of the search tree is defined by (7) with each $C_t^s(j)$ obtained by removing from $C_0^s(j)$ the channels that have been assigned to other users along the partial path l_t that ends at the current node.*

Obviously, the node value so defined is a function of the node and the associated partial path. We thus denote the node value by $v(l_t)$ to emphasize such a dependence. To see that the node value is indeed a lower bound, we first notice that, like the cost function of the DPRA algorithm, it is a function of a channel allocation that is illegitimate and optimistic. The channel allocation is illegitimate because a subcarrier may be assigned to more than one user and it is optimistic since an user tends to own more than its share of subcarriers, resulting in reduced required power. In the search tree shown in Fig. 1, each parent node has d child nodes to enclose all possible solutions. When we search along a path to visit an internal node of the tree, we compute the associated "node value" by (7) with each $C_t^s(j)$ obtained by removing from $C_0^s(j)$ the channels that have been assigned to other users along the partial path l_t from the root node to the current node. Obviously, the node value so defined is a function of the node and the associated partial path. To emphasize such a dependence, we denote the node value by $v(l_t)$. As an user's SCS is a decreasing function of the partial path length in the sense that a child node's SCS is a subset of their parent node's, the node value of a child node must be equal to or greater than that of its parent node. In other words, the fact

$$C_0^s(j) = C(j) \supset C_1^s(j) \supset C_2^s(j) \supset \cdots \supset C_N^s(j),\ \forall\ j \tag{3.10}$$

implies

Property 1. *Both $g(R_j, C_t^s(j))$ and the cost function $J_t(\mathbf{C}_t^s)$ defined by (3.7) are increasing functions of t.*

As every complete path is associated with a sequence of shrinking SCSs $\{C_0^s(j), C_1^s(j), \cdots C_N^s(j)\}$ and $C_N^s(j)$ is the cost of this path, we have

Property 2. *The node value $v(l_t)$ defined by (7) is a lower bound for the cost of any complete path that coincides with the t-level partial path l_t.*

Thus, if J_t at a parent node is not smaller than the upper bound, we are sure that there is no optimal solution in its child nodes and one should check other nodes of the same level. On the other hand, the order of visiting the d child nodes of a parent node should be based on their node values as the node value represent our current best estimate of all subsequent assignments. For the convenience of subsequent reference, we summarize these two observations, which often brings about significant search complexity reduction (see Table 3.3) of a B&B-based resource allocation (BBRA) algorithm, as

Guideline 1. *The order of visiting d child nodes of a given parent node should be the same as the ascending order of the magnitudes of the corresponding node values. In other words, one should visit the node with the least node value, followed by the one with second smallest node value, and so on.*

Guideline 2. *When visiting a node (say at the tth level) of a partial path l_t, we compute the node value $v(l_t)$ and compare it with the current upper bound B_u. If $v(l_t) < B_u$ then visit its first child node in the next level. Otherwise, searching on the subtree rooted at this node is terminated and the search should continue on the next unvisited child node of the same level or backtrack to the next unexplored nodes in the previous level, where the order of d siblings descending from the same parent node is determined by Guideline 1.*

Because only a complete path corresponds to a candidate solution, the depth-first-search (DFS) strategy is suitable for our B&B approach. The initial upper bound B_u^0 can be obtained by the DPRA algorithm. The ensuing DFS searching procedure tries to continuously separate the parent space into the subproblem (child) space. Therefore, we have

Guideline 3. *Upon arriving at the final level, we check the resulting cost (node value) to see if B_u^t has to be updated. We then backtrack to the nearest parent node determined by Guideline 1 and resume the searching process.*

Note that the above three *Guidelines* are valid for general B&B approaches and are listed for the convenience of subsequent discussions.

Definition 1 and the above guidelines all assume that we compute the node values when transversing along a path based on the same principle used by the DPRA algorithm. In other words, every user is given all channels initially and, at each level along a path, a channel is assigned to the user associated with the selected child node and removed from all other users' SCSs. Such a procedure will not exclude any legitimate solution from the tree search. With this assumption, we note that the node values along a path may reach a steady state before the leaf node is visited. A necessary condition is

Property 3. *Further traversing on a path will not change the node value if the set of remaining unassigned channels $\bigcap_j C_t^s(j) = C_U$ satisfy either (i) $\forall\, i \in C_U \Rightarrow r_{ij} = 0, \forall\, j$,* *or (ii)* $\forall\ i \in C_U, i \in C_t^s(j)$ *for only one* j.

This property can be used to accelerate our search without missing the optimal solution.

Guideline 4. *Besides those terminations specified by Guideline 2, early termination (of a path) is possible if one of the conditions in Property 3 is satisfied.*

Since computing the node value via (7) requires repeated calls to the OMPA subroutine, the search complexity can be reduced if we can minimize the numbers of calls. A careful examination of (7) and the search procedure reveals

Guideline 5. *In computing the node value for the* k*th child node of a* $(t-1)$*th level parent node, the fact* $C_t^s(k) = C_{t-1}^s(k)$ *implies that* $g(R_k, C_t^s(k;k)) = g(R_k, C_{t-1}^s(k))$. *Furthermore, if in computing the parent node's value we have* $r_{tj} = 0$ *for some* j, *then* $g(R_j, C_t^s(j)) = g(R_j, C_{t-1}^s(j))$. *For both cases there is no need to call the OMPA subroutine to compute the minimum required power. Finally, although for a fixed* k, d *OMPA calls are needed in computing each cost* $g(R_j, C_t^s(j;k))$, $d-1$ *of them can be reused for other* k*'s.*

The last two guidelines can be used to reduce the computing complexity of the DPRA algorithm as well. In particular, *Guideline 5* implies that only d OMPA calls are needed to compute d child node values of a given parent node.

3.3.2 Sorting The Serving Channels

We have suggested a channel ordering for the DPRA algorithm according to the maximum GNR's. This channel indexing is simple but, according to our simulation, does not yield fast convergence. Like the DPRA algorithm, the order of the channels is very important. If our ordering (indexing) of the channels is such that the ith ($i < N$) channel is so "bad" that it is not used in the final optimal solution (no user really wants it) then we have to check all its d child nodes in the next level. Simulations indicate that the channel ordering affects the search speed significantly. In view of *Guideline 1*, *Property 2* and given we have decided the first k channels, the $(k+1)$th channel should be the most demanded one such that its assignment to a user (thus is removed from the SCSs of all other users) increases the costs (node values) of all other users most significantly. The channel-sorting algorithm based on this idea, is presented in Table 3.4.

We have several remarks on the above channel-sorting process.

R1. The sole purpose of this algorithm is channel-sorting and the corresponding channel assignments are auxiliary operations, not to be realized.

R2. Step 2 in Table 3.4 defines the most demanding channel as the one that offers the highest sum rate and is requested by two or more users given the current SCS collection. When a channel offers the highest rate but serves only one user, it must render relatively low GNR for all other users, hence the decision of its order in the tree should be postponed.

R3. Step 4 deals with the ordering of those channels which, after several rounds of filtering the most demanded channels, are still requested by one user only.

With this channel-sorting procedure and in view of the properties and guidelines mentioned before along with our definition of the node value, we propose the BBRA algorithm of Table 3.2.

3.3.3 Complexity Reduction Techniques

To explore the effectiveness of various techniques implied by properties and guidelines on reducing the computing complexity, we have performed 10^6 simulated runs of the BBRA algorithm that incorporates (1) the channel-sorting process in Table 3.4 and the combinations of (2) *Guideline 1*, (3) *Property 3* and (4) the fifth Guideline. The numbers of users and channels are *5* and *128*, respectively, and the normalized rate for each user

Step 1:	(Initialization) Use the DPRA algorithm to obtain the initial upper bound B_u^0 and the channel-sorting process in Table 3.4, to rearrange the channel order. Set the initial level at $\aleph = 1$
Step 2:	Visit the child nodes of the $\aleph$th level according to *Guideline 1* and invoke *Guideline 2*. Set $\aleph \leftarrow \aleph + 1$ if no backtracking is needed; otherwise set $\aleph \leftarrow \aleph - 1$. *Property 3* should be used at every node visited to check the possibility of early termination of a candidate path.
Step 3:	Go to Step 2 if $\aleph < N$. If $\aleph = N$ then terminate the searching process if all nodes have been visited or been excluded from further consideration; otherwise invoke *Guideline 3*. Set $\aleph \leftarrow \aleph - 1$ and go to Step 2.

Table 3.2: A branch and bound based resource allocation (BBRA) algorithm

$d = 5, N = 128$	DPRA	(1)	(1)+(2)	(1)+(3)	(1)+(4)	(1)+(2)+(3)+(4)
$E[n_{op}]$	44.61	2587.2	773.98	578.09	116.19	88.32
$E[n_{op} \mid n_{op} < 2 \times 10^5]$	44.61	1717.8	773.98	549.91	93.78	88.32
$\text{Prob}[n_{op} > 2 \times 10^5]$	0	0.0012	0	0.00005	0.00004	0
$\max\{n_{op}\}$	81	21622894	180132	1800602	981053	587

Table 3.3: The effects of (1) channel-sorting in Table 3.4, (2) Guideline 1, (3) Property 3/Guideline 4, and (4) Guideline 5 on the computing complexity reduction of the BBRA algorithm; 10^6 runs are performed to obtain the statistics. The complexity is measured in terms of numbers of calls n_{op} to the OMPA algorithm. The complexity of the DPRA algorithm is also included for comparison purpose.

is uniformly distributed in $[0, 3]$. The results are summarized in Table 3.3 with the complexity measured in terms of the number of calls to the OMPA algorithm.

Channel-sorting is most critical for with other conventional channel-sorting methods (e.g., that used by DPRA), the searching complexity often becomes greater than 10^6. Hence it is always assumed as part of the initialization step in the BBRA algorithm. The reuse of existing OMPA results (i.e., *Guideline 5*) also brings about significant reduction as it is applicable in every node visit. Proper branching and early terminations help accelerating the search process a lot as well.

3.3.4 Application to Integer Constellation Systems

With minor modifications, our algorithms remain valid and are applicable for solving a similar RA problem with integer constellation (discrete-rate) constraints. All we to have to do is inserting an SNR gap, which depends on the constellation size and the BER requirement, in the rate-power equation (2) and replacing the OMPA (water-filling procedure) algorithm by a known bit-loading algorithm, e.g., Campello's optimal algorithm whose complexity is upper-bounded by $O(N)$ [3].

A B&B approach was also suggested in [13] to solve a similar problem for integer constellation systems. Besides not having the attributes mentioned in the second paragraph of this section, their method differs from ours in at least two major aspects. First, their approach implies a tree structure that grows a $(dM + 1)$-ary sub-tree out of each node where M is the number of discrete rates allowed while we need only a d-ary sub-tree. In other words, [13] converts both user and rate selections into node selections, each node represents a fixed user/rate assignment for a given subcarrier but our tree search has to do with user selection only. Second, each node value (lower bound) of [13] is obtained by solving a linear programming problem after relaxing three major constraints, namely, (i) the SUPS, (ii) the single-rate-per-subcarrier, and (iii) the discrete-rate constraints. The first two relaxations are directly related to their tree structure and the last relaxation is needed to convert the integer linear programming problem into a (real) linear one which is much easier to solve. As a result, the lower bound so obtained is not very tight. In contrast, we use either the OMPA algorithm or Campello's algorithm [3] to perform the corresponding (provisional) optimal rate/power allocation once an user is selected (for using a subcarrier). The corresponding bounds do not have to remove constraints (ii) and (iii) mentioned above whence are much tighter and result in far less search complexity.

3.4 Numerical Results and Algorithmic Complexity

We report some simulated performance and complexities of the proposed algorithms and two suboptimal algorithms modified from existing ones in this section. As the performance of two proposed algorithms is almost identical, that of the BBRA algorithm is not shown and is used as the reference for comparison only. 10^5 runs, each with a different channel realization, are performed to obtain the numerical results presented here.

3.4.1 Relative Efficiency Performance

▶ **Performance of BBRA and DPRA algorithms**

As only the GNRs a_i affect the performance we assume, without loss of generality, that $\sigma_{ij} = \sigma$, $\forall\, i, j$. We normalize the bandwidth of each sub-carrier (channel) such that $W = 1$ and set the normalized noise power lever σ^2 to be 1. We also normalize the Rayleigh-distributed channel gains $||h_{ij}||^2$ such that $E[|h_{ij}|^2] = 1$ and assume that channels are independently faded. These two normalization assumptions effectively imply $E[a_{ij}] = 0$ dB. Since the channel capacity or the normalized rate is a function of the product $p_{ij}a_{ij}$, the simulation results shown in Figs. 3–6 are scalable in the sense that a higher (lower) $E[a_{ij}]$ needs a proportionally lower (higher) minimum required power. The normalization of the channel bandwidth W has a similar purpose in interpreting the normalized data rates R_i which now have the unit of bits/sec/Hz. We also have the normalized sum rate as $\sum_{j=1}^{d} R_j$. Various normalized rate distributions with the same sum rate are examined.

Let J_{DP} and J_{BB} be the total required transmit power determined by the DPRA and BBRA algorithms, respectively, and define the relative efficiency (RE) of the former algorithm by

$$\eta = 1 - \frac{E[J_{DP}] - E[J_{BB}]}{E[J_{BB}]} \tag{3.11}$$

Since the power-rate allocation is solely determined by OMPA once the subcarrier assignment is fixed, we say two algorithms give the same solution if both suggest the same subcarrier allocation. Fig. 3.2 plots the probability that the DPRA algorithm converges to the optimal solution for several cases ($N = 64, 128$ $d = 5, 10, 15$). Since the BBRA algorithm is guaranteed to give the optimal solution, this probability is equal to $\mathrm{P}_r[J_{DP} = J_{BB}]$. It is found that when $d \ll N$ (say $d/N < 0.1$) the probability that the DPRA algorithm yields the optimal solution is greater than 0.9 if the sum rate is less than 8 bits/sec/Hz. Although for other cases under investigation, this probability is smaller than 0.9, Fig. 3.3, which plots the RE of the DPRA algorithm, indicates that the corresponding solutions still lie very close to the optimal one. It is clear that the DPRA algorithm is capable of offering a near-optimal solution that even in the worst case (64 channels, 15 users and a normalized required sum rate of 20) it achieves a RE as high as 99.82%.

In general, the larger the number of the users d or the sum rate is, the less efficient the DPRA algorithm becomes. Such a behavior is consistent with the fact that, when d increases but N is fixed, a correct channel selection at each level become less likely so is the probability of obtaining the optimal channel allocation. On the other hand, the assumption of independent fading of channels implies that the probability of having "good" channels increases as N increases, and the probability of correct or good decision at each level increases as well. Hence, for a fixed d, the probability of obtaining the optimal or near optimal solution is an increasing function of N and so is the RE (η).

Step 1:	(Initialization) Set $t = 1$ and let the SCS and the assigned channel set (ACS) for user j be $C_t^s(j) = \{1, 2, \cdots, N\}$ and $C_t^a(j) = \emptyset$, respectively.
Step 2:	(Find the most demanded channel) Compute the sum rate $R_i^s \stackrel{def}{=} \sum_{j=1}^{d} r_{ij}$ for each channel, where r_{ij} is obtained by applying the OMPA algorithm for each SCS $C_t^s(j)$, compute $\Psi(i) = \{\ j \mid 1 \le j \le d,\ r_{ij} \ne 0\}, C_A = \bigcup_{j=1}^{d} C_t^a(j)$, $C_h = \{\ i \mid 1 \le i \le N,\ \lvert\Psi(i)\rvert \ge 2\}$ If $\lvert C_h\rvert \ne \emptyset$ and $\ell = \max_{i \in C_h} R_i^s$, then the ℓth channel is re-indexed as channel t (i.e., $\mu(\ell) = t$) and assign this channel to user k if $k = \max_j r_{\ell j}$. Go to Step 4 if $\lvert C_h\rvert = \emptyset$.
Step 3:	(Updating) The ACS for user k and the SCSs are updated by $C_t^a(k) \leftarrow C_t^a(k) \cup \{\ell\}$, $C_t^s(j) \leftarrow C_t^s(j) \setminus \{\ell\}, \forall\ j \ne k$, respectively. The tree level index is updated by $t \leftarrow t + 1$. If $t < N$, go to Step 2; otherwise, the sorting process is completed.
Step 4:	(Sorting the less demanded channels) If $C_t^s(j) \bigcap C_t^a(j) = \emptyset, \forall\ j$, go to Step 5; otherwise, for all j, $C_t^s(j) \bigcap C_t^a(j) \ne \emptyset$, modify the corresponding SCS by $C_t^s(j) \leftarrow C_t^s(j) \setminus \{j_m\}$, where $j_m = \arg\max_{i \in C_t^s(j) \bigcap C_t^a(j)} r_{ij}$, and go to Step 2.
Step 5:	(Sorting the remaining channels) The order (numbering) of the channels in the set $\{i \mid 1 \le i \le N, i \notin C_A\}$ is determined by the maximum GNR criterion used in the DPRA algorithm.

Table 3.4: The channel-sorting algorithm

► **Performance of Representative Sub-optimal Algorithms**

Although many RA schemes have been proposed, they assume different scenarios and costs. Those dealing with RA problems similar to (4) often follow a three-step procedure [5],[6]. (S1) *Resource allocation*–determine the resource (number of channels) to be given to each user based on some criterion. (S2) *Subcarrier assignment*–decide which subcarrier should serve which user. (S3) *Local optimization*–each user computes the optimal power allocation according to its channel set and rate requirement. The average GNR criterion, i.e., the BABS (bandwidth assignment based on GNR) algorithm [5], is perhaps the simplest and most popular choice for use in (S1). Such an approach treats the channel of concern as a flat-faded wideband channel when determining the number of subcarriers an user is entitled to possess. It simplifies the resource allocation procedure by ignoring the selectivity of a wideband channel but is very likely to exclude the optimal solution from further consideration. Many methods were proposed to obtain the water-filling solution for (S3) with various degrees of precision. The main difference lies in (S2).

The first approach called the amplitude craving greedy (ACG) algorithm [5] sequentially assigns the subcarriers to the user with the largest GNR unless the channel number quota determined in (S1) has been exceeded. An alternate method called rate craving greedy (RCG) algorithm [5] finds the water-filling rate level for all users, assuming they have been given all subcarriers. The subcarriers are then assigned to the one with the highest achievable rate unless its channel number quota is exceeded. The original ACG and RCG algorithms can not be used to solve (4) as they are designed for discrete-rate constraints. Moreover, they use an approximate instead of exact water-filling solution. For the purpose of fair comparison, we modify both algorithms by using the rate-power equation (2) and the OMPA solution. The resulting algorithms are henceforth referred to as the modified ACG (MACG) and RCG (MRCG) algorithms, respectively.

We assume GNR =20 dB for all subcarriers and $R_j = 5$ bits/sec/Hz for all j. Besides the independent-fading channel model, for $N = 128$ we also consider the ITU Vehicular A model [11] which has been adopted by UMTS and WiMax forum as one of the reference channel models. The RE performance shown in Fig. 3.4 indicates that our DPRA algorithm does outperform both MACG and MRCG algorithms. It yields a near-optimal solution that even in the worst case ($N = 64, d = 14$), achieves a 99.92% RE while the the two modified algorithms give 91.74% and 92.92% efficiencies. Due to the fixed rate requirement, a larger N results in better performance for all suboptimal approaches. The increase of d, on the other hand, leads to reduced efficiency but DPRA is much more robust in the sense of maintaining almost constance RE for different d, N and channel conditions. The optimal allocation probabilities for these two suboptimal algorithms are not presented for they are simply too small.

3.4.2 Complexity Evaluation

The computing complexity of various RA methods is dominated the number of calls to the single-user (mono-rate) water-filling algorithm whether it is the OMPA or any other algorithm that computes the power or rate associated with each channel assigned to an user. An exhaustive search requires $O(Nd^N)$ single-tone power- or rate-level computing operations [4] or $O(d \cdot d^N)$ calls of OMPA.

► **Average complexities of the proposed algorithms**

For the DPRA algorithm, at most $d^2 \times N$ calls of OMPA is needed. But the complexity of BBRA method is difficult to analyze directly for it depends on the channel order and the initial upper bound value. By using computer simulation, we estimate the average complexities, measured in terms of number of calls of the OMPA algorithm, of the DPRA and BBRA algorithms and present the results in Figs. 3.5-3.6. We assume that the normalized GNR is 0 dB and examine the required complexity for different numbers of users with the same normalized sum rate. A few observations on the last two figures can be made. First, because of *Guidelines 4* and 5, the complexity of DPRA algorithm is limited to at most $dN + 2d$ OMPA calls. Second, although the complexities of both DPRA and BBRA algorithms increase with the number of users d, the latter is much more sensitive to this parameter. Finally, the average complexity of the BBRA algorithm is higher when there are 64 channels than if there are 128 channels. The reason for this interesting fact is that there are more good channels when $N = 128$ and, for a fixed sum rate requirement, as good channels tend to support higher data rates, fewer channels are needed and early terminations due to *Guideline 2* and *Guideline 4* occur more often.

► Complexities of Other Sub-optimal Algorithms

Since the complexity of water-filling is a function of the channel number involved, a more precise and fair comparison is counting the number of rate(power)-evaluation iterations, i.e. the [MR2]-[MR4] loop of the OMPA algorithm; see Appendix A. For BBRA or DPRA algorithms, the iteration number in every call is upper-bounded by $\log_2 N$ as bisection search is in place. The complexity of the DPRA algorithms can be reduced by using *Guidelines 4, 5* and the iteration number is thus upper-bounded by $(dN + 2d)\log_2 N$.

Fig. 3.7 shows the average complexity performance of our algorithms and the MACG, MRCG algorithms for $N = 64$ and 128. The system and channel parameter values used here are the same as those used in Fig. 3.4. As expected, the performance in correlated fading is worse than that in independent fading and the computation complexity of all algorithms are far less than the DRRA upper-bound, $(dN + 2d)\log_2 N$. Moreover, BBRA requires the highest average complexity, followed by DPRA, MRCG and the MACG algorithms. The complexity of the DPRA algorithm is about twice that of the RCG based algorithm but is far less than that of the BBRA algorithm. The DPRA algorithm, as mentioned before, yields near-optimal performance and is robust against the variations of the numbers of users and subcarriers.

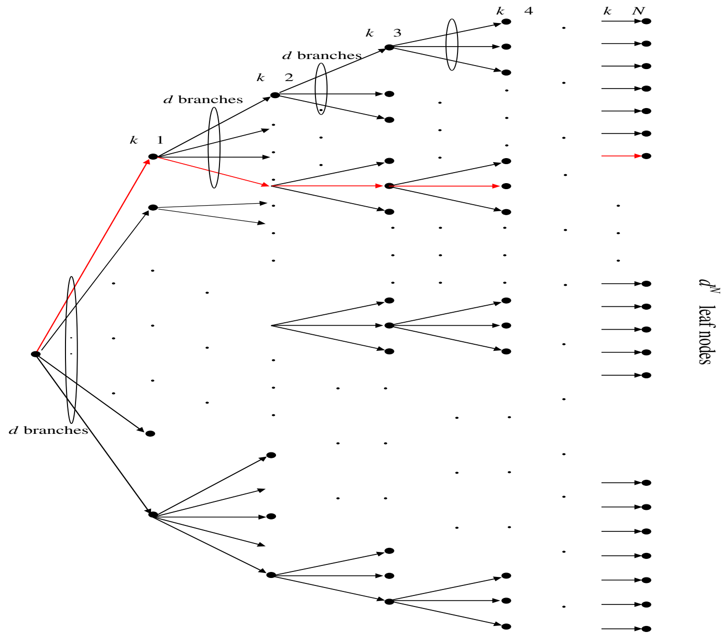

Figure 3.1: A complete search tree for multiuser channel allocation. For the DPRA algorithm, only one child node survives at each level.

3.5 Chapter Summary

OFDMA is an effective multiple access scheme in a wideband wireless mobile network. Besides its anti-fading capability, an OFDMA system can achieve high spectral efficiency in a multiuser environment by adaptively allocating subcarriers and time slots to the the most suitable users with the minimum required transmit power. An efficient dynamic RA algorithm to solve the corresponding constrained optimization problem in real time is thus crucial for realizing this potential advantage.

Based on the principles of dynamic programming and branch-and-bound, we propose two algorithms–the DPRA and BBRA algorithms–which give either near-optimal or optimal solution. In contrast to the existing algorithms, which suffer from the shortcomings of requiring high complexity and/or unsatisfactory performance, the DPRA algorithm renders near-optimal performance with relative low complexity. Since the existing efficient algorithms are designed with a discrete-rate constraint and use some suboptimal

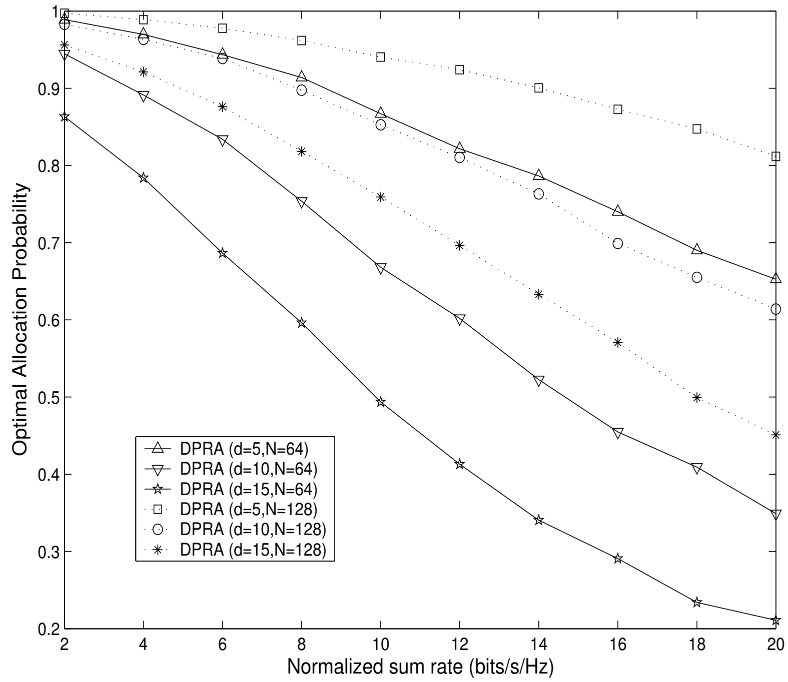

Figure 3.2: The DPRA algorithm's probability of correct convergence (i.e., the probability of obtaining the optimum subcarrier/power/rate allocation) in an OFDMA downlink.

water-filling solution, we make some modifications for fair comparisons. As expected, the resulting ACG and RCG based DPRA algorithms are shown to provide less satisfactory performance with reduced complexities. With proper reuse of the water-filling solution obtained in earlier stages, the average DPRA complexity can be further reduced and is insensitive to d, N and the required sum rate. The average complexity of the BBRA algorithm, on the other hand, is at least an order higher than that of the DPRA algorithm when the number of users is greater than 10 but is still much less than the known algorithms for obtaining the optimal solution.

Our numerical experiment in both independent and correlated fading environments have demonstrated that the near-optimal DPRA algorithm is suitable for real-time resource allocation application and the optimal BBRA algorithm is practical only if $d \leq 5$. Nevertheless, the latter algorithm offers the optimal solution and performance for large N and d with reasonable complexity, which has never been achieved before and is needed

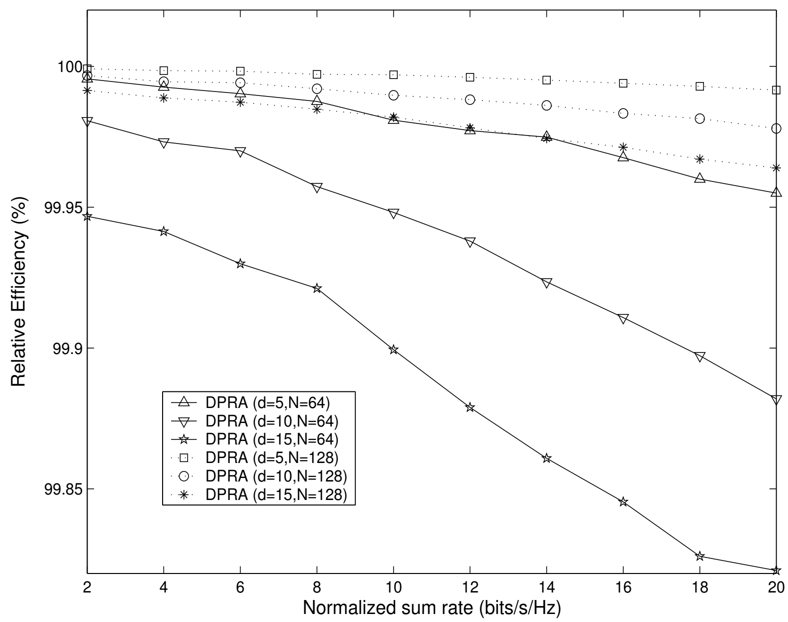

Figure 3.3: Average relative efficiency (η) performance of the DPRA algorithm.

for benchmarking and comparison purposes. Finally, we would like to mention that, although we restrict our discourse to the capacity (rate) constraint, the proposed algorithms are applicable to other constraints such as fixed BER or weighted capacity constraint by modifying the corresponding rate-power function.

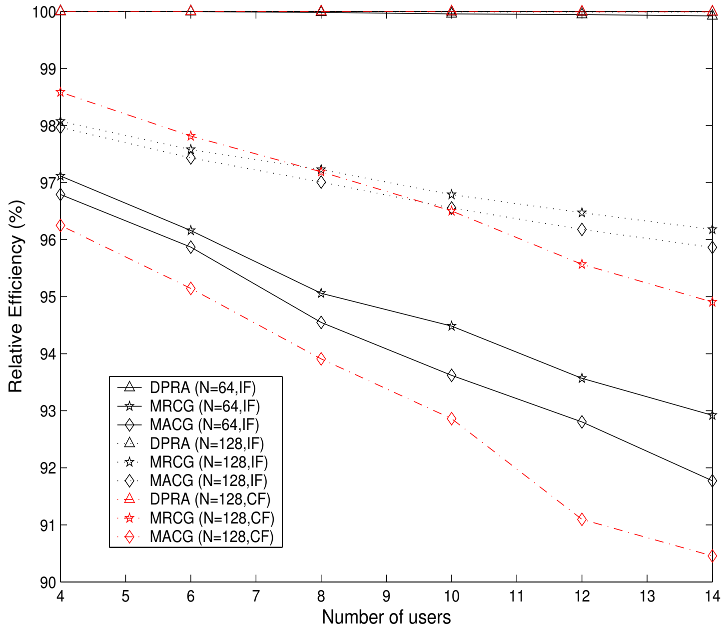

Figure 3.4: Average relative efficiency of the DPRA, MDPRA, and MRCG algorithms; $R_j = 5$ bps/Hz for all j, IF = independent fading, CF = correlated fading.

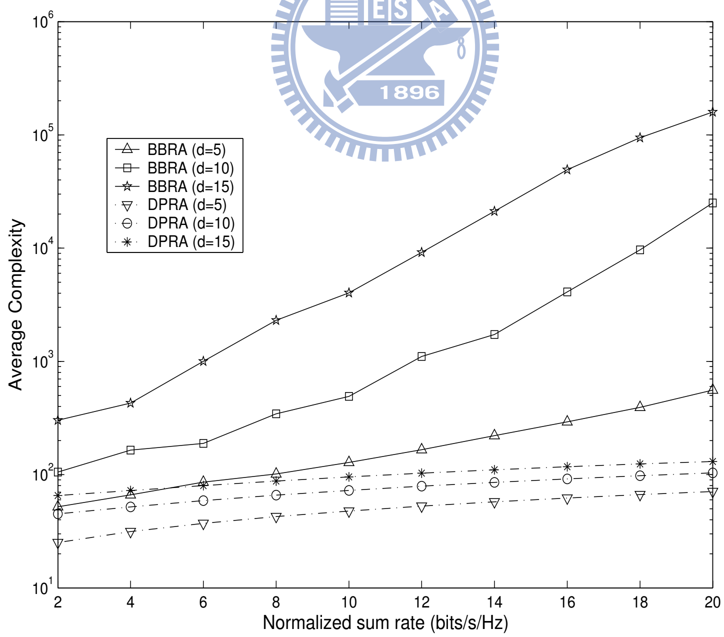

Figure 3.5: Average complexities (numbers of calls to the OMPA algorithm) for the BBRA and DPRA algorithms in a 64-subcarrier OFDMA system.

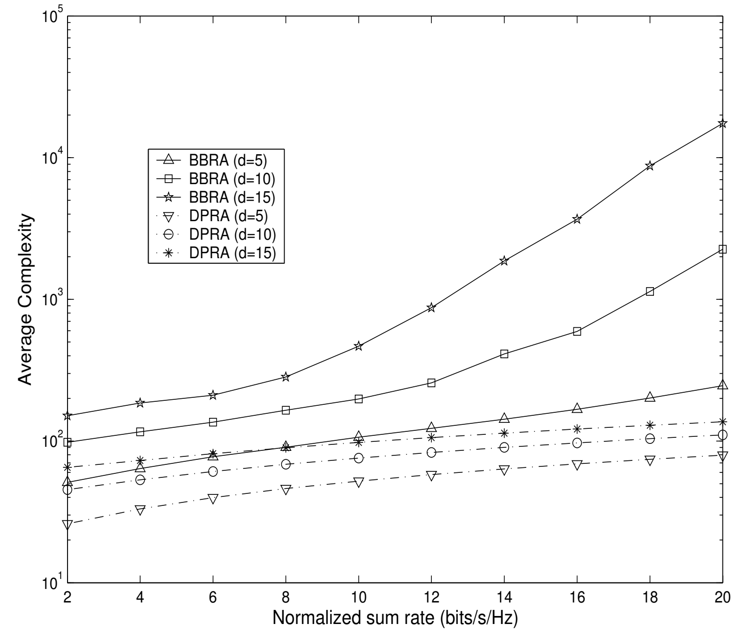

Figure 3.6: Average complexities (numbers of calls to the OMPA algorithm) for the BBRA and DPRA algorithms in a 128-subcarrier OFDMA system.

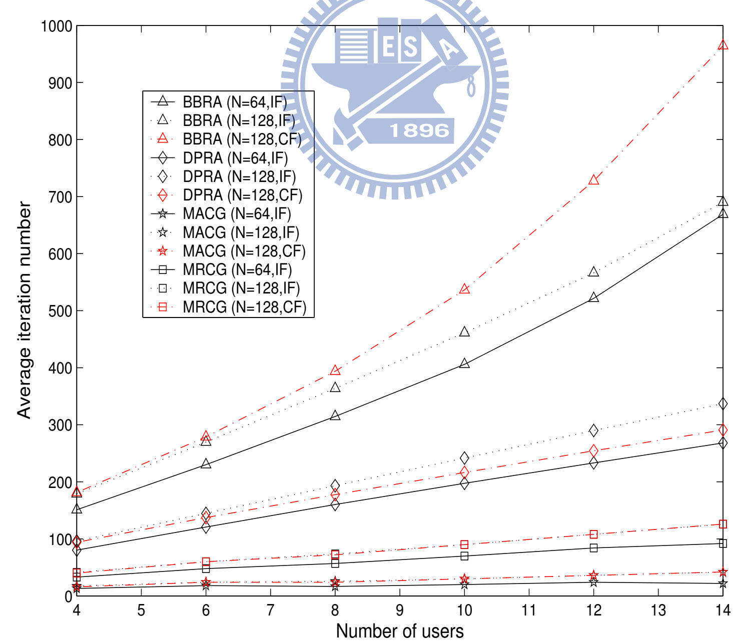

Figure 3.7: Average number of power (or rate) level evaluation iterations for various dynamic RA algorithms; $R_j = 5$ bps/Hz for all j, IF = independent fading, CF = correlated fading.

Chapter 4

Capacity Maximization Resource Algorithms

Recall that the object of the fundamental problem–single user resource-allocation across parallel orthogonal channels–is to maximize the achievable sum rate subject to a power constraint. This problem, as mentioned before, can be optimally solved by means of the water-filling method and an efficient algorithm, i.e., the OMPA algorithm has been developed and given in Appendix A. The rate allocation in each subcarrier is then determined by the corresponding power allocation through the rate-power equation. In a single-cell multiuser system, when subcarriers assignment is fixed, optimal power-rate allocation can be independently carried out via the OMPA algorithm. However, since a subcarrier's link gain depends on the user involved, which leads to the so-called multiuser diversity, subcarrier and power assignment must be jointly considered. We thus have a scenario similar to that addressed in the previous chapter.

4.1 System Description

We consider a single-cell multiuser OFDMA uplink system. The users want to access the base station and must share the same multicarrier band. We assume that the base station and each user are equipped with a single antenna and the inter-symbol interference (ISI) is completely removed by maintaining the orthogonality amongst subcarriers through proper netwrok timing synchronization so that each subcarrier suffers from frequency-flat fading. In addition, we also assume the multiple channels are slow-fading and all channel gains remain unchanged within each OFDM frame. The base station is provided by mobile users the perfect channel gain information via reliable feedback channels.

The jth user's transmission rate R_j is given by

$$R_j = \sum_{i=1}^{N} A_{j,i} r_{j,i},$$

where $r_{j,i}$ denotes the jth user's transmission rate over the ith subcarrier. Define the subcarrier assignment matrix $[A]_{ji} = A_{j,i}$, where the (j,i)th entry, $A_{j,i}$, is nonzero and equal to one iff the ith subcarrier has been allocated to the ith user. The rate-power function is the same as (3.2).

4.2 Resource Allocation Algorithm for Weighted Sum Rate Maximization

The weighted sum rate maximization (WSRmax) problem of concern can be stated as

$$\begin{aligned}
\text{maximize} \quad & \sum_{j=1}^{K} \mu_j \sum_{i=1}^{N} A_{j,i} r_{j,i} \\
\text{subject to} \quad & \sum_{i=1}^{N} A_{j,i} p_{j,i} \le P_j, \quad \text{for } 1 \le j \le K \\
& \sum_{j=1}^{K} A_{j,i} \le 1, \ \text{for } 1 \le i \le N \\
& A_{j,i} \in \{0,1\}, \ \text{for } 1 \le i \le N, 1 \le j \le K
\end{aligned} \tag{4.1}$$

where μ_j is the weight assigned to data rate of user j and K denotes the number of users. Given the weight vector and the channel gains, we want to find the power allocation that maximizes the weighted sum rate with the power constraints $\{P_j\}$. This is not a convex optimization problem due to the subcarrier set selection procedure must be carried out. However, The boundary of the achievable rate region can be traced by solving this problem for all possible weight vectors.

If all users have equal weights, the original problem becomes a sum-rate maximization problem. In addition, if the individual power constraint is replaced by the total sum power constraint for downlink scenario, the optimal subcarrier set selection becomes that of assigning each subcarrier to the user with the highest subcarrier GNR and the optimal power allocation is a single level water-filling over this optimal set. Once, the users' weights are not equal, giving the subcarrier to the user with the highest subcarrier gain is not necessarily the optimal solution.

4.2.1 A Dynamic Programming Based Algorithm

Since the channel condition for a specific subcarrier may looks "good" to more than one user, there are competitions among users. Conventional suboptimal algorithms tend to assign a subcarrier to the user who has the largest GNR for that subcarrier and give a fixed pre-determined number of subcarriers to each user to avoid the unfair situation that a few users dominate the competition result and get most of the subcarriers. Unfortunately, for our case it is not easy to determine the number of channel assigned to each user. In general, the above conventional subcarrier assignment scheme is not a good solution for maximizing either the weighed sum rare or sum rate. First, each user has a peak transmitted power limit, hence if it is given too many subcarriers there will be not enough power to achieve the maximum rate. Second, those users who lost competition and are given few subcarriers with small GNR can not use their power efficiently. In short, the system throughput (sum rate) performance can be improved by other allocation methods.

The proposed constraint relaxation and reinstallation with user removal process has been proved to be both powerful and efficient for this kind of optimization problem. As has been shown in Chapter 3, the DPRA algorithm based on this concept does yield near-optimal solution. We thus use the same approach for dealing with the current WSRmax non-convex problem. The proposed subcarrier allocation process still consists of a series (N-level) of users deletion decisions. Similar to that described in Section 3.2, we assign a subcarrier to an user at each level and simultaneously remove this subcarrier from the SCS of other users. As the objective function is the weighted sum rate instead of the total transmitted power, the selection of the child node at each level is based on the new cost function given by

$$\begin{aligned} J_t(\mathbf{C}_t^s) &= \max_{1\le k\le K} \left\{ \sum_{j=1}^{K} \mu_j \hat{g}(P_j, C_t^s(j;k)\ ) - \omega_t \left[\sum_{j=1}^{K} \delta(N-t-|C_t^s(j;k)|) \right] \right\} \\ &\overset{def}{=} \max_{1\le k\le K} J_t^k(\mathbf{C}_t^s) \end{aligned} \tag{4.2}$$

where $\hat{g}(P_j, C_t^s(j:k))$ represents the maximal rate of user j based on its virtual channel set $C_t^s(j:k)$. We use an algorithm similar to the OMPA algorithm of Appendix A to compute the maximal transmission rate for the single user case.

The DP-based WSRmax solution can be described by using 4.15 to replace the ones of Table 4.1 in Chapter 3.

4.2.2 A Low Complexity Algorithm via Dual Decomposition

The Lagrangian of the WSRmax problem can be defined over domain $\mathcal{D}$ as

$$\mathcal{L}\left(\{p_{j,i}\},\{r_{j,i}\},\lambda\right) = \sum_{j=1}^{K} \mu_j \sum_{i=1}^{N} r_{j,i} - \sum_{j=1}^{K} \lambda_j \left(\sum_{i}^{N} p_{j,i} - P_j \right), \tag{4.3}$$

where the domain $\mathcal{D}$ is defined as the set of all non-negative $p_{j,i}$ for $j = 1, 2, \cdots, K$, and $i = 1, 2, \cdots, N$ such that for each i at most only $p_{j,i}$ is positive for $j = 1, 2, \cdots, K$ for satisfying the constraint $\sum_{j=1}^{K} A_{j,i} \le 1$. $\lambda = (\lambda_i, \lambda_2, \cdots, \lambda_K)$ represents the Lagrangian multipliers of all users.

Then, the Lagrange dual function can be given

$$g(\lambda) = \max_{\{p_{j,i}\},\{r_{j,i}\}\in\mathcal{D}} \mathcal{L}(\{p_{j,i}\},\{r_{j,i}\},\lambda). \tag{4.4}$$

The maximization of $\mathcal{L}$ can be decomposed into the following N independent optimization problems, which is given by

$$\tilde{g}_i(\lambda) = \max_{\{p_{j,i}\},\{r_{j,i}\}\in\mathcal{D}} \left\{ \sum_{j=1}^{K} (\mu_j r_{j,i} - \lambda_j p_{j,i}) \right\}, \quad \text{for } i = 1, 2, \cdots, N. \tag{4.5}$$

Hence, the original Lagrange dual function can be written as

$$g(\lambda) = \sum_{i=1}^{N} \tilde{g}_i(\lambda) + \sum_{j=1}^{K} \lambda_k P_j. \tag{4.6}$$

With a fixed λ, the maximization of 4.5 is a convex function of $\{p_{j,i}\}$. We take the derivative of this object regarding $p_{j,i}$, the optimality condition to maximize $\tilde{g}_i(\lambda)$ is given by

$$p^*_{j,i}(\lambda_j) = \left(\frac{\mu_j}{\ln 2\lambda_j} - \frac{1}{a_{j,i}} \right)^+ \tag{4.7}$$

where $a_{j,i}$ denotes the GNR of the jth user's ith subcarrier. Because each subcarrier can serve at most one user, $\tilde{g}_i(\lambda)$ can be obtained as

$$\tilde{g}_i(\lambda) = \max_j \left\{ \mu_j r^*_{j,i}(\lambda_j) - \lambda_j p^*_{j,i}(\lambda_j) \right\}, \text{ for } i = 1, 2, \cdots, N. \tag{4.8}$$

Finally, we need to find the optimal λ to maximize $g(\lambda)$. The update of λ can be done by using the ellipsoid method until every user power converges. Unfortunately the corresponding complexity for finding the optimal solution is still very high. Hence we propose a simple and efficient dual decomposition based iterative algorithm to obtain a suboptimal solution.

For the dual decomposition approach, if λ is fixed, we can choose the best user for each subcarrier by (4.8). Since λ can gave as large as K^N possible values, we try to search for a suboptimal λ and obtain the corresponding feasible subcarrier assignment within at most N iterations. First of all, we release all channel-using constraint such that the maximal number of each sucarriers is enlarged to be the number of total users and each user perfectly detect its data without interference from other users who use the same subcarrier. Hence, each user can use the single user rate maximization algorithm, which can be obtained by modifying the OMPA algorithm, to compute every user's subcarrier efficiencies as following.

$$\epsilon^{(k)}_{j,i} = \mu_j \tilde{r}^{(k)}_{j,i} - \lambda^{(k)}_j \tilde{p}^{(k)}_{j,i} \quad \text{for} \quad 0 \le k \le N-1 \tag{4.9}$$

where $\lambda_j^{(k)}$ denotes the Lagrangian multiplier of user j at the kth iteration (level) and

$$\tilde{p}_{j,i}^{(k)} = \left(\frac{\mu_j}{\ln 2 \lambda_j^{(k)}} - \frac{1}{a_{j,i}} \right)^+ \quad (4.10)$$

$$\tilde{r}_{j,i}^{(k)} = \log_2 \left(1 + \tilde{p}_{j,i}^{(k)} a_{j,i} \right) \quad (4.11)$$

At each iteration, $\mu_j^{(k)}$ and $p_{j,i}^{(k)}$ will be re-computed by the single user rate maximization algorithm since the SCS of the jth user may have been changed. In other words, each iteration will decide the assignment of one subcarrier means that the communication over this subcarrier has to be realistic considered by forcing the number of users at most be one.

$$\nu_i = \sum_{j=1}^{K} \epsilon_{j,i} \quad (4.12)$$

where ν_i denotes the overall efficiency value of the ith subcarrier, the corresponding individual efficiency values are computed by (4.9). The larger ν_i is, the more efficiency the ith subcarrier is for all users. Hence, the subcarrier with larger ν need to be assigned as soon as possible such that the user who lost it at current iteration can enlarge efficiency values of other unassigned subcarrier according to its requirement (rate) at next iteration. Once the subcarrier number is obtained at each iteration, the remaining work is to decide which users will be given this subcarrier. For the SISO FDMA system, just only one user can occupy the subcarrier of each iteration. Hence, we pick up the user with highest efficiency over this subcarrier. Hence, within each iteration, a smart subcarrier selection and a corresponding user selection are operated according on the efficiency value computed by dual decomposition. Finally, after N iterations, a suboptimal feasible solution can be obtained since all subcarriers's constraints are satisfied. The detailed algorithm is summarized in the following table.

4.2.3 Numerical Results and Discussions

In this section, we show some simulated performance of weighted sum rate maximization algorithms. For simplicity, we set all users' weight to one. For the MaxGain algorithm, a

Step 1:	(Initial channel allocation)
	Set $C(j) = \{i \mid 1 \le i \le N\}$, for $1 \le j \le K$.
	set $S = \{1, 2, \cdots, N\}$ and $t = 0$
Step 2:	(channel selection)
	each user individually run OMPA algorithm based on its available channel set $C(j)$
	if $t < N$
	$i^* = \arg\max_{i \in S} \nu_i$
	Set $S = S \backslash \{i^*\}$, $t = t + 1$, then goto Step 3.
	else goto Step 5.
	end
Step 3:	(user election)decide which user can transmit in the i^* subcarrier
	and other users are deleted
	$j^* = \arg\max_{1 \le j \le K} \epsilon_{j,i}$.
	goto Step 4
Step 4:	(modify channel set $C(j)$ for $1 \le j \le K$.)
	for $j = 1 : K$
	if $j \ne j^*$ **then** $C(j) \backslash \{i^*\}$ **end**
	end
	goto Step 2
Step 5:	(Output) The final channel allocation and user election

Table 4.1: A resource allocation algorithm for weight sum rate maximization.

subcarrier is always assigned to the user with the largest GNR. In Figs. 4.1 and 4.2, we plot the sum rate performance and average fairness indices as a function of the number of users. We assume that the maximum individual normalized transmitted power is P and all subcarriers suffer from independent Rayleigh fading. The fairness index is defined as

$$F = \frac{\left(\sum_{j=1}^{K} R_j\right)^2}{K\left(\sum_{j=1}^{K} R_j^2\right)} \tag{4.13}$$

The sum rate performance of the proposed and the MaxGain algorithms are fairly close as is shown in Fig. 4.1. MaxGain algorithm's fairness performance, however, is much inferior to the proposed algorithms, especially when the number of users becomes large.

In Fig. 4.3, we consider the ITU Vehicular A model [11] which has been adopted

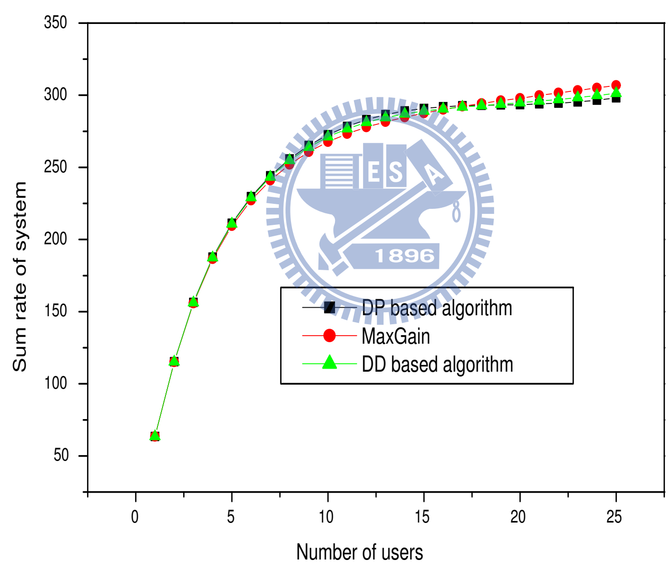

Figure 4.1: Average sum rate performance as a function of number of users; $P = 60$, $N = 64$, GNR $= 0$ dB.

by UMTS and WiMax forum as one of the reference channel models. We observe that our algorithms achieve far better sum rate performance than that of the MaxGain algorithm. This is a result of the clustered behavior of the correlated fading channels,

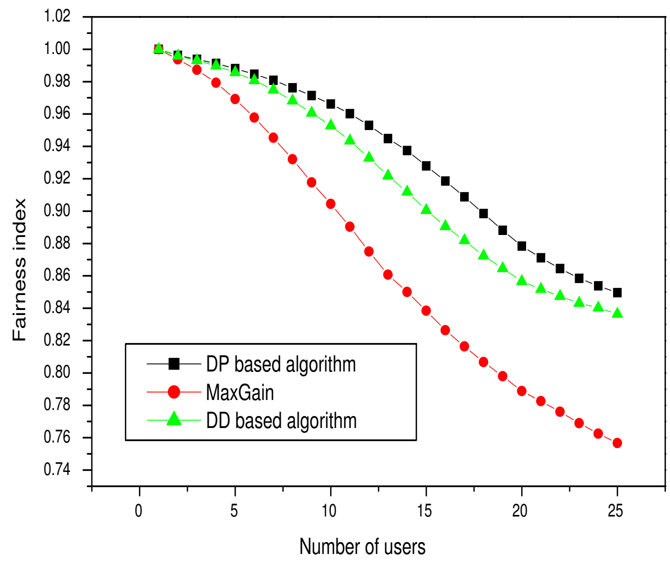

Figure 4.2: Average fairness index performance as a function of number of users; $P = 60, N = 64$, GNR = 0 dB.

i.e., neighboring subcarriers tend to have similar gains and thus users who have a strong (weak) subcarrier are likely to have a bunch of strong (weak) subcarriers. The clustered behavior results in dominance by some users, i.e., a great portion of the subcarriers is given to a small group of users while some users get only a few weak subcarriers. It leads not only to the "waste" of good subcarriers because of peak power limitation but also to the "waste" of power in subcarriers with poor gains. Fig. 4.3 shows the system throughput improvement defined as

$$\frac{\text{Sum Rate}_{New} - \text{Sum Rate}_{MaxGain}}{\text{Sum Rate}_{MaxGain}}$$

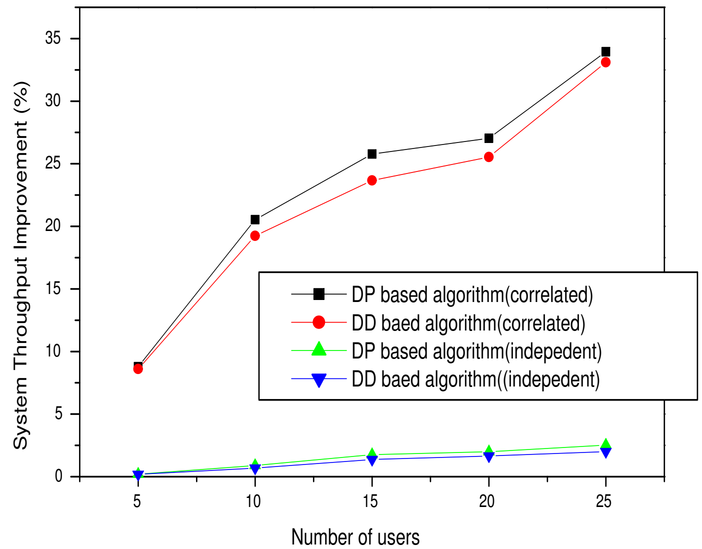

Figure 4.3: System throughput improvement versus number of users in a correlated fading channel; $N = 128$, and GNR = 0 dB.

4.3 Resource Allocation Algorithms for Product Rate Maximization

In this section, we will take proportional fairness into account in resource allocation management. Most of the previous approaches show how to efficiently maximize total transmission or minimize the total transmitted power under the related constraints of system and users. In addition to the issue that to minimize total transmitted power under all users' constraints can ensure the benefits of users far away from the base station, most formulated problems and the corresponding solutions are focus on the efficiency issue such that the users closer to the base station or with higher power capability will get the most resource/benefits. Although the max-min criterion has been considered for the resource allocation fairness among users in multiuser orthogonal frequency-division multiplexing (OFDM) systems. However, there are two main drawbacks in the max-min criterion. First, resource allocation based on the max-min criterion is not easy to take into account that users' different requirements. Secondly, since the max-min approach deals with the worst-case scenario, the system will penalize user with better channels and drop the efficiency very much. Based on the above discussion, it is necessary to develop an approach that jointly considers the resource allocation fairness and system efficiency. In addition, it is also low-complexity for practical scenario.

4.3.1 Nash Bargaining Solution and Problem Formulation

In this subsection, we briefly introduce the concept of Nash bargaining solution. The Nash bargaining solution from game theory is a classical solution to fairness resource allocation problem.

We denote by $\mathbf{K} = \{1, 2, \cdots, K\}$ the set of players and by $\mathcal{U}$ a closed and convex subset of $\mathbb{R}^K$ which represents the set of feasible payoff allocations that the players would obtain if they cooperate. We further denote by R^j_{min} the minimal payoff the jth player (user) expects; otherwise, it will not cooperate and define $\mathbf{R}_{min} = (R^1_{min}, \cdots, R^K_{min})$. Suppose $\{R_j \in \mathcal{U} | R_j > R^j_{min}\ \forall j \in \mathbf{K}\}$ is a nonempty bounded set, then the pair $(\mathcal{U}, \mathbf{R}_{min})$ is called a K-person bargaining problem. The notion of Pareto optimal is defined as a selection criterion for the bargaining solutions.

Definition 2. *The point $(R_1, R_2, \cdots, R_K)$ is said to be* **Pareto optimal***, if and only if there is no other allocation $\hat{R}_j \geq R_j, \forall j$, which yields superior performance for some specific users without incurring performance degradation for some other users.*

As there may exist many Pareto optimal solutions if no additional condition is imposed. By introducing some fairness requirements, we have a fair Pareto optimal operating point called the Nash bargaining solution (NBS), which is defined below.

Definition 3. *$\bar{\mathbf{r}}$ is said to be an* **NBS** *in $\mathcal{U}$ for $\mathbf{R}_{min}$, i.e., $\bar{\mathbf{r}} = \phi(\mathcal{U}, \mathbf{R}_{min})$, if the following axioms are satisfied.*

(1) Individual Rationality: $\bar{R}_i = \sum_{j=1}^{N} \bar{r}_{ij} \geq R^i_{min}\ \forall i$.

(2) Feasibility: $\bar{\mathbf{r}} \in \mathcal{U}$

(3) Pareto Optimality: For every $\hat{\mathbf{r}} \in \mathcal{U}$*, if* $\sum_{j=1}^{N} \hat{r}_{ij} \geq \sum_{j=1}^{N} \bar{r}_{ij}\ \forall i$*, then* $\sum_{j=1}^{N} \hat{r}_{ij} = \sum_{j=1}^{N} \bar{r}_{ij}$

(4) Independence of Irrelevant Alternatives: If $\bar{\mathbf{r}} \in \mathcal{U}' \subset \mathcal{U}$*,* $\bar{\mathbf{r}} = \phi(\mathcal{U}, \mathbf{R}_{min})$*, then* $\bar{\mathbf{r}} = \phi(\mathcal{U}', \mathbf{R}_{min})$

(5) Independence of Linear Transformations: For any linear scale transformation ψ*,* $\psi(\phi(\mathcal{U}, \mathbf{R}_{min})) = \phi(\psi(\mathcal{U}), \psi(\mathbf{R}_{min}))$.

(6) Symmetry: If $\mathcal{U}$ *is invariant under all exchanges of agents,* $\phi_j(\mathcal{U}, \mathbf{R}_{min}) = \phi_{j'}(\mathcal{U}, \mathbf{R}_{min})$ $\forall j, j'$.

It can be shown that there exits an unique NBS [15] which happens to be equivalent to the maximum product rate solution.

Note that a popular fairness criterion is the max-min criterion which implies that the system performance will be dominated by the user with the worst channel condition. This criterion actually penalizes the users with good subcarriers, and as a result, generates inferior overall performance. This is not truly fair for users with more or better communication opportunities. Another fairness index is the proportional fairness concept, which can be shown to be a special case of NBS fairness. As a result, by maximizing the product rate we in effect impose the fairness concern in our algorithm. Our RA problem can now be stated as follows.

$$\begin{aligned}
\text{maximize} \quad & \prod_{j=1}^{K} \sum_{i=1}^{N} A_{j,i} r_{j,i} \\
\text{subject to} \quad & \sum_{i=1}^{N} A_{j,i} p_{j,i} \leq P_j, \quad \text{for } 1 \leq j \leq K \\
& \sum_{j=1}^{K} A_{j,i} \leq 1, \ \text{for } 1 \leq i \leq N \\
& A_{j,i} \in \{0, 1\}, \ \text{for } 1 \leq i \leq N, 1 \leq j \leq K
\end{aligned} \tag{4.14}$$

In the next section we propose a suboptimal dynamic programming based solution for product rate maximization.

4.3.2 DP Based Algorithm for Product Rate Maximization

The DP based resource allocation algorithms developed in Chapter 3 and previous section have been proved to be capable of offering near-optimal performance. Using the same idea, we modify the corresponding cost function J_t as

$$\begin{aligned} J_t(\mathbf{C}_t^s) &= \max_{1\le k\le d} \left\{ \prod_{j=1}^{d} \hat{g}(P_j, C_t^s(j;k)\ ) - \omega_t \left[\sum_{j=1}^{d} \delta(N-t-|C_t^s(j;k)|) \right] \right\} \\ &\stackrel{def}{=} \max_{1\le k\le d} J_t^k(\mathbf{C}_t^s) \end{aligned} \tag{4.15}$$

where $\hat{g}(P_j, C_t^s(j:k))$ represents the maximal rate of user j based on its virtual channel set $C_t^s(j:k)$. We use a modified algorithm similar to the OMPA algorithm to compute the maximal transmission rate for the single user case.

The DP-based WSRmax solution can be describe by using 4.15 to replace the ones of Table 3-1 in chapter 3.

4.3.3 Numerical Results and Discussions

Here, we take the MaxGain algorithm discussed in the last section and the DPRA algorithm under maxmin criterion for comparison. The second algorithm is denoted by DP based maxmin algorithm which uses DPRA process to maximum the minimal user rate at each level. In the other words, in DP based maxmin algorithm we just exploit the following objective function to replace the original one.

$$\begin{aligned} J_t(\mathbf{C}_t^s) &= \max_{1\le k\le K} \left\{ \min_{1\le j\le K} \hat{g}(P_j, C_t^s(j;k)\ ) - \omega_t \left[\sum_{j=1}^{K} \delta(N-t-|C_t^s(j;k)|) \right] \right\} \\ &\stackrel{def}{=} \max_{1\le k\le K} J_t^k(\mathbf{C}_t^s) \end{aligned} \tag{4.16}$$

In Fig. 4.4 under considering independent channel fading scenario, we can find the solution of DP based NBS has the largest fairness index. It is expected that DP based maxmin algorithm outperforms than the MaxGain algorithm.

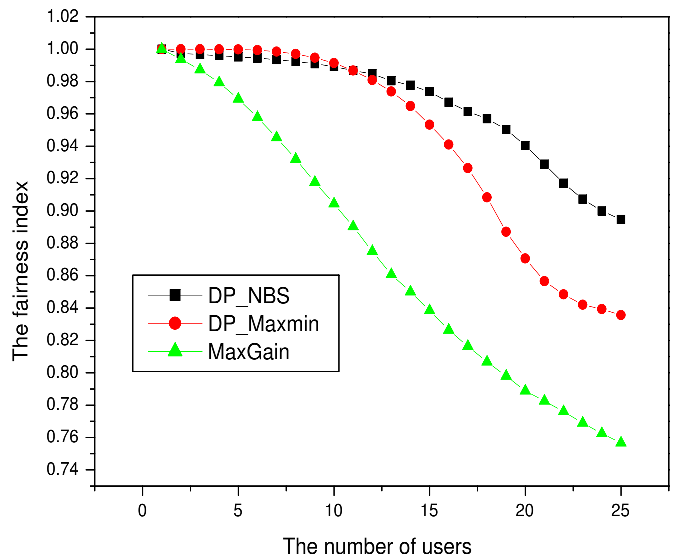

Figure 4.4: Fairness index performance as a function of number of users; $N = 64$, $P = 60$, GNR = 0 dB.

Fig. 4.5 shows the performance in correlated channels. We assume the same the correlated channel model as that mentioned in the last section. From Figs. 4.4 and 4.5, we find that, for independent fading channels, fairness can be enhanced by adding the number of subcarriers. Simultaneously, it is shown in Fig. 4.5 that there is no noticeable difference in fairness performance between DP based NBS algorithm and the DP based maxmin algorithm. But in Fig. 4.5, the DPRA algorithm for NBS still yield the best system performance measured by the sum rate without degrading too much fairness performance.

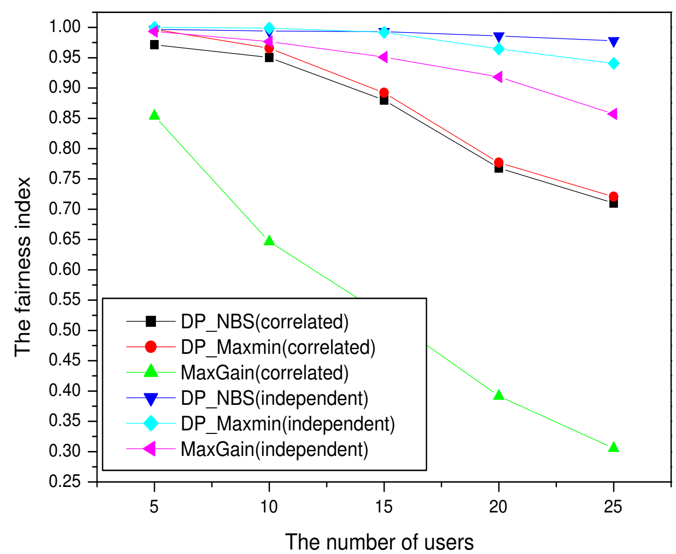

Figure 4.5: Fairness index versus number of users, ($N = 128$, GNR = 0 dB) for correlated and independent fading channels.

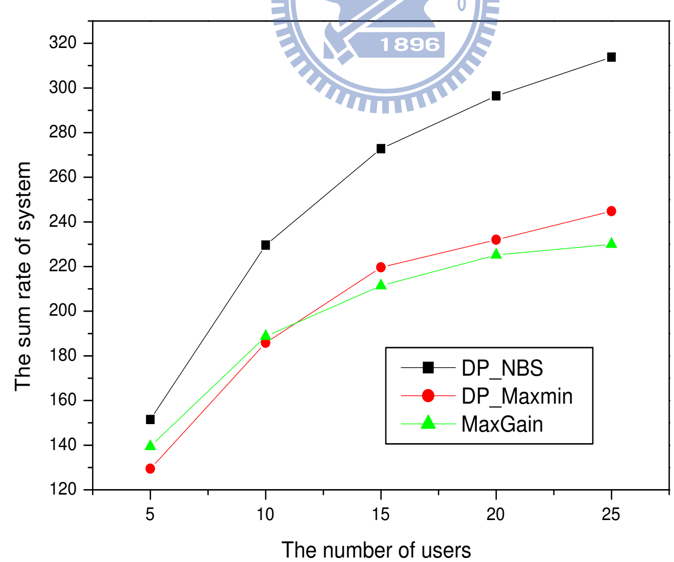

Figure 4.6: Sum rate performance versus number of users in a correlated channel; $N = 128$, GNR = 0 dB.

Chapter 5

Resource Allocation for MIMO-OFDMA Downlinks

In the downlink of a multiuser multiple-input multiple-output (MIMO) system, the multiple antennas at the base station (BS) allow for spatial multiplexing of transmissions to multiple users in the same time slot and frequency band. Due to their geographical dispersion, coordination among users is difficult, which makes the downlink of a multiuser system more challenging compared to single user MIMO systems.

From information theoretic point of view, the sum capacity achieving precoding or preprocessing technique is dirty paper coding (DPC) [18]. The DPC solution uses successive interference pre-cancellation approaches that employ complex encoding and decoding schemes. An intensive research effort is thus underway to devise suboptimal but more practical approaches to multiuser downlink signal processing. Beamforming or transmit pre-processing is a suboptimal and reduced complexity (compared to DPC) strategy, where each user stream is coded independently and multiplied by a beamforming weight vector for transmission through multiple antennas.

Proper design of the beamforming weight vectors allows the interference among different streams to be minimized (or eliminated), thereby supporting multiple users simultaneously. As a result, multiuser MIMO system substantially increases the system capacity by multiplexing users in the spatial domain.

The challenge is thus to design transmit and receive processing vectors such that space-division multiplexing is effectively achieved. Despite its suboptimality, for independently fading channels linear beamforming has been shown to achieve the best trade-off between performance and complexity. In this chapter, we consider orthogonal linear precoding techniques to achieve orthogonal space division multiplexing (OSDM) in the downlink of multiuser MIMO systems, in which both base station as well as mobile stations employ multiple antennas. The orthogonal precoding allows transmission to the mobile users to be multiuser interference free. OSDM for multiuser MIMO systems has been proposed by several researchers [19] [20]. We consider two OSDM techniques that use a subspace projection methods to design precoding matrices: block diagonalization. With block diagonalization, each user's precoding matrix is designed such that the transmitted signal of that user lies in the null space of all other remaining users' channels, and hence multiuser interference is pre-eliminated. With the sum power constraint, block diagonalization takes the sum rate maximization approach, which tends to select the strong users often causing unfairness among users. Hence, minimizing transmit power while achieving desired quality of service for users may be interesting in practice. In this thesis, we will propose a low-complexity suboptimal algorithm to solve this problem. Due to the null space dimensional requirements of block diagonalization, the numbers of users supported in the same time/frequency slot are limited for a given number of transmit antennas. Therefore this technique should be combined with radio resource management so that a multiuser diversity gain can be achieved. Multiuser diversity arises when a large number of users with independently fading channels are present, and hence it is likely that a user or multiple users experience high channel gain in any given time/frequency slot. In addition ,to reduce the hardware complexity at the mobile units, there is no receive antenna selection when users are selected for transmission in any given time/frequency slot, with which all antennas are selected for reception.

In [21], the authors proposed a user selection for single carrier case in MISO systems. However, due to the coupling of all users' channels in an OFDMA network, optimal selection of user subsets involves exhaustive search through all combinations of active users, which is computationally very complex for systems with a large number of users and frequency/time slots. Hence, we propose simplified user selection resource allocation algorithm for block diagonalization. The rest of the Chapter is organized as follows. Section 2.1 describes the system model. The brief review of block diagonalization technique is shown in section 5.2. Section 5.3 describes the problem formulation. Finally, we propose our low-complexity suboptimal algorithm and some numerical results for discussion.

5.1 System Description

In this section, we provide a general description of a typical cellular-based MIMO-OFDMA system. Fig. 5.1 shows a downlink multiuser MIMO system in which a base station transmits data to K users over N subcarriers. The BS is equipped with N_T antennas and the jth user terminal has n_j antennas. The total number of receive antennas is thus given by $N_R = \sum_{j=1}^{K} n_j$. We also assume that BS has perfect channel state information (CSI) of all active system users. The CSI may be obtained via channel reciprocity for time division duplex (TDD) systems or through feedback links. The issue of inaccuracy of CSI at the transmitter as well as the emerging partial CSI feedback schemes are some of the practically important issues for multiuser MIMO downlink transmission but will not be addressed.

We denote by $\mathbf{H}_{j,i} \in \mathbb{C}^{n_j \times N_T}$ the downlink channel matrix of the jth user. A flay Rayleigh fading channel is assumed so that the elements of $\mathbf{H}_{j,i}$, $j = 1, 2, \cdots, K$ and $i = 1, 2, \cdots, N$ can be modelled as independent and identically distributed (i.i.d.) complex zero-mean Gaussian random variables with variance of 0.5 per dimension. Hence, for the ith subcarrier, the received signal of the jth user can be expressed as

$$\mathbf{y}_{j,i} = \mathbf{H}_{j,i} \sum_{k=1}^{K_i} \mathbf{W}_{k,i} \mathbf{x}_{k,i} + \mathbf{n}_{j,i} \quad \text{for} \quad \forall j \in \Omega_i \tag{5.1}$$

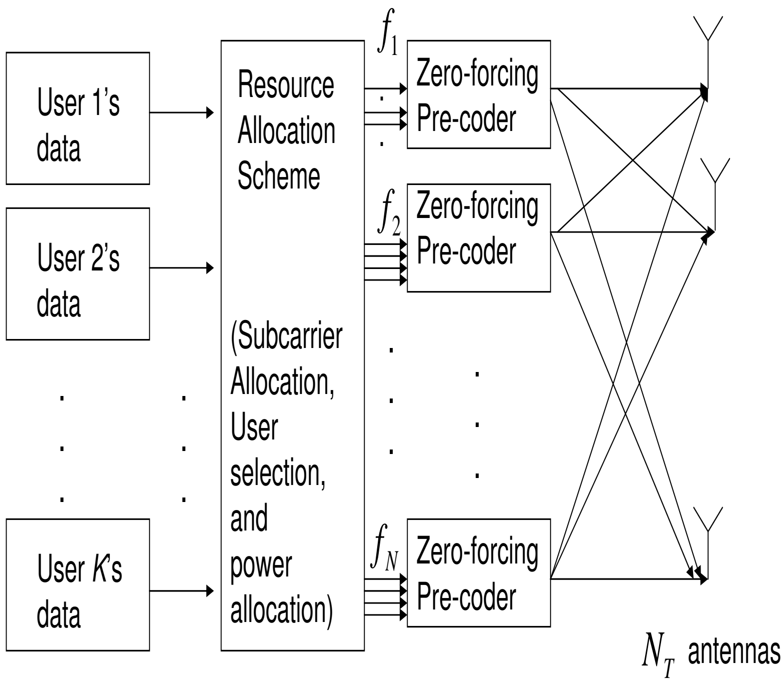

Figure 5.1: Block diagram of a multiuser MIMO system.

where $\mathbf{n}_{j,i} \in \mathbb{C}^{n_j \times N_T}$ denotes zero mean additive Gaussian noise with $E\{n_{j,i}n_{j,i}^H\} = \sigma^2\mathbf{I}_{n_j}$. $K_i = |\Omega_i|$ represents the number of users who simultaneously communicate over the ith subcarrier. After linear processing at the receiver, the received signal can be reformulated as

$$\hat{\mathbf{y}}_{j,i} = \mathbf{U_{j,i}}^H(\mathbf{H}_{j,i}\sum_{k=1}^{K_i}\mathbf{W}_{k,i}\mathbf{x}_{k,i} + \mathbf{n}_{j,i}) \tag{5.2}$$

where $\mathbf{U_{j,i}} \in \mathbb{C}^{N_T \times n_i}$ is the receive process matrix of user j over the ith subcarrier.

We focus on the scenario of multiuser interference free environments. Based on the above assumptions, $\mathbf{W}_{k,i}$ should be designed such that

$$\mathbf{H}_{j,i}\mathbf{W}_{k,i} = 0 \text{ for all } \ k \neq j \text{ and } \ k, j \in \Omega_i \tag{5.3}$$

and

$$\sum_{j=1}^{K_i} n_j \leq N_T, \text{for } \ \forall j \in \Omega_i, \quad \text{and } 1 \leq i \leq N \tag{5.4}$$

As a result, the multiuser MIMO system with K_i users over the ith subcarrier can be decomposed into K_i parallel single-user MIMO system as follows.

$$\mathbf{y}_{j,i} = \mathbf{H}_{j,i}\mathbf{W}_{k,i}\mathbf{x}_{k,i} + \mathbf{n}_{j,i} \tag{5.5}$$

5.2 The Block Diagonalization Approach

In this section we describe the transmission scheme for the MIMO-OFDM channels by using linear transmit and receive equalization to block-diagonalize the spatial channel. Block diagonalization (BD) can be used to spatially separate the users when more than one user share a certain subcarrier. This process creates decoupled spatial channels for all the users when they use the same subcarrier. For each user, singular-value decomposition (SVD) is applied to the combined channel matrix of all the other users. The last few right singular vectors that correspond to zero singular values form a base for the null space of this combined matrix. Next, each user's channel matrix is multiplied by the corresponding null space base that was obtained earlier, and SVD is performed on the resulting matrix. These two steps would give the transmit and receive equalization matrices and different user's MIMO channels become completely decoupled, with no interference amongst users.

We use an example to explain the above discussion, assuming that there are K_m users transmitting their data in subcarrier m simultaneously.

Let us denote the aggregate channel and precoding matrices of all users by

$$H = (H_1^T \ H_2^T \ \cdots \ H_{K_m}^T)^T \tag{5.6}$$

$$W = (W_1 \ W_2 \ \cdots \ W_{K_m}). \tag{5.7}$$

The precoding matrices of users are designed using null space projection method as follows. Let us define a $(\sum_{j=1,j\neq k}^{K_m} n_j) \times N_T$ aggregate channel interference matrix for user k as

$$\tilde{H}_k = (H_1^T \ \cdots H_{k-1}^T H_{k+1}^T \ \cdots \ H_{K_m}^T)^T \tag{5.8}$$

Zero multiuser interference constraint requires that the precoding matrix W_k of user k lies in the null space of $\tilde{H}_k$, which requires the null space of $\tilde{H}_k$ to have a dimension greater than zero. This condition imposes a constraint on the number of base station antennas to be

$$N_T \geq \sum_{j=1}^{K_m} n_j, \tag{5.9}$$

We assume that the channel matrices are of full rank for all users (which occurs with probability of one in i.i.d. Gaussian channels). We denote the SVD of $\tilde{H}_k$ by

$$\tilde{H}_k = \tilde{U}_k \begin{pmatrix} \tilde{\Sigma}_k & \mathbf{0} \end{pmatrix} (\tilde{V}_k^1 \quad \tilde{V}_k^0)^H \tag{5.10}$$

where $\tilde{\Sigma}_k$ is a $\tilde{r}_k \times \tilde{r}_k$ diagonal matrix containing $\tilde{r}_k$ non-zero singular values of $\tilde{H}_k$, and $\tilde{r}_k$ is the rank of $\tilde{H}_k$.

$\tilde{V}_k^0$ has the $N_T - \tilde{r}_k$ right singular values of $\tilde{H}_k$ as its columns. These columns constitute the orthonormal basis for the null space of $\tilde{H}_k$. We assume that the fading among the antennas of a user as well as among the users is independent (hence the matrices are of full rank). Hence, there exists $N_k = N_T - \tilde{r}_k$ columns of $\tilde{V}_k^0$, which form the null-space basis of $\tilde{H}_k$. Constructing the precoding matrix with the columns of $\tilde{V}_k^0$ will satisfy the zero multiuser interference constraint. With this, the multiuser channel decouples into K parallel non-interfering single-user MIMO channels (also referred to as null space projected channels), which is expressed as

$$\hat{H}_k = H_k \tilde{V}_k^0 = \hat{U}_k \hat{S}_k \hat{V}_k^H \tag{5.11}$$

where the last equality represents the SVD of $\hat{H}_k$. Thus, on each subcarrier, for the group of users that are currently served, their channels are completely decoupled, and they do not interfere each other.

5.3 Problem Formulation and the Optimal Solution

In this section, we present a mathematical formulation for the problem of total power minimization under user rate constraints and derive the optimal solution. The complexity of the optimal solution is huge because of the need for an exhaustive search over a large set of possible subcarrier allocation options.

Our objective is to find the optimal subcarrier allocation $\sigma_{j,i}$ and power allocation $p_{j,i}$ that minimize the overall transmit power subject to each user's data rate requirement R_j in bits per second (bps), which can be formly stated as

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{N}\sum_{j=1}^{K} p_{j,i} \\ \text{subject to} \quad & \sum_{i=1}^{N} r_{j,i} \geq R_j \quad \forall j \\ & p_{j,i} \geq 0, \quad \forall i, j \end{aligned} \tag{5.12}$$

where $r_{j,i}$ denotes the transmitted rate of user j on subcarrier i, it can be written as

$$r_{j,i} = \sum_{l=1}^{\eta_{j,i}} \log_2 \left(1 + \frac{\tilde{p}_{j,i,l} s_{j,i,l}^2}{\Gamma N_0}\right) \tag{5.13}$$

where $s_{j,i,l}$ is the lth diagonal element of user j's equivalent channel $\hat{S}_{j,i}$ on subcarrier i and $0 \leq \eta_{j,i} \leq \min(n_j, N_T), \forall i$.

In addition, $s_{j,i,l}$ is dependent on the user selection $\sigma_{j,i}$ on subcarrier i, which indicates the presence of the jth user on subcarrier i. If the jth user is present, $\sigma_{j,i} = 1$; otherwise, $\sigma_{j,i} = 0$. $\{\sigma_{j,i}\}$ denotes the user selection on each subcarrier. $p_{j,i,l}$ is the power loading on eigenchannel l for user j on the i subcarrier, and $p_{j,i} = \sum_{l=1}^{\eta_{j,i}} \tilde{p}_{j,i,l}$. According to the previous definition of $\sigma_{j,i}$, if $\sigma_{j,i} = 0$, $p_{j,i}$ will be set to 0, $\tilde{p}_{j,i,l} = 0, \forall l$,and $r_{j,i} = 0$. In (5.13), Γ represents the SNR gap, which is given by

$$\Gamma = -\frac{5 \text{ BER}}{1.5} \tag{5.14}$$

When the subcarrier assignment $\{\sigma_{j,i}\}$ is fixed, the power allocation can be found separately for each user. For example, if user k is of interest, the original problem can be written as

$$\begin{aligned}
\min_{\{p_{j,i}\}} \quad & \sum_{i=1}^{N} p_{j,i} \\
\text{s.t.} \quad & \sum_{i=1}^{N} r_{j,i} \geq R_j \\
& p_{j,i} \geq 0, \forall i \\
& p_{j,i} = 0, \text{ if } \sigma_{j,i} = 0.
\end{aligned} \tag{5.15}$$

Through water-filling processes, the corresponding optimal power and rate allocation can be carried out over user j's eigenchannels across all the subcarrier, which can given by

$$\tilde{p}_{j,i,l} = (\frac{\mu_j}{\ln 2} - \frac{\Gamma N_0}{s_{j,i,l}^2})^+ \tag{5.16}$$

$$\tilde{r}_{j,i,l} = \log_2 \left(1 + \frac{\tilde{p}_{j,i,l} s_{j,i,l}^2}{\Gamma N_0}\right) \tag{5.17}$$

In a water filling process, $\mu_j/ln2$ can be interpreted as the common water level of the power or water that is poured over channels, with the river beds being equal to $\Gamma N_0/s_{j,i,l}^2$. The conventional method for obtaining optimal water level is to start with the maximum number of streams, $\frac{\mu_j}{ln2}$ is evaluated for a decreasing number of streams until the point where the water level is above the highest river bed.

However, in order to obtain a globally optimal solution, an exhaustive search is needed over all possible subcarrier assignments $\{\sigma_{j,i}\}$ to find the minimum transmit sum power. Thus, K water-filling procedures over Nn_k singular values have to be carried out for each of 2^{KN} possibilities. Even if a constraint is imposed such that only one user can occupy each subcarrier, there still be K^N possibilities to be tested.

5.4 A Low-complexity Power-Minimization Solution

The transmitter performs user selection and linear beamforming so that each subcarrier can support multiple users while meeting the users' rate constraints. Since finding the optimal user set and multiple subcarriers assignment to minimize the total used power requires an almost exhaustive search and is not computationally feasible.

In this section, an efficient solution to the power minimization problem is derived based on a Lagrange dual decomposition. The overall process to obtain the solution is similar to the DPRA algorithm. All users will share all subcarriers initially. Then, remove unsuitable users at each subcarrier iteratively in order to satisfy the constraint that the number served by a subcarrier is at most equal to the maximum number of users. The maximum number of users over each subcarrier is designed by the precoding scheme used. We adopts the block diagonalization approach to ensure that there will be no cochannel interference. The maximum number of users served by a subcarrier is therefore equal to N_T/n_r, where the number of receive antennas at each user terminal is assumed to be $n_j = n_r$ for all $1 \leq j \leq K$.

There are two major differences between current proposal and the DPRA algorithm of Chapter 3. First, the channel order is not needed since at each level we determine the assignment of the subcarrier whose efficient value computed by dual decomposition is the largest among all remaining subcarries which are still in the SCS of all users. Second, user selection is replaced by an simple method based on the utilization efficiency computed by dual decomposition; there is no need to compute the K possible cost values. However, the DPRA algorithm still can be adopted to solve the MIMO OFDMA resource allocation problem by extending the number of tests at each level (subcarrier) from the number of users (K) to all possible user selection. But the resulting complexity would be so high that it becomes impractical.

We now describe the proposed solution as following. Our algorithm starts, like previous cases, by releasing all channel-using constraint such that the maximal number of each sucarriers is increased to be the number of total users, assuming that each user can perfectly detect its data without interference from other users who use the same subcarrier. Simultaneously, each user employs the OMPA algorithm by extending the original N subcarriers to $N \times r$ eigenchannels. We then compute the efficiency of all eigenchannels of each user over all subcarriers by

$$\epsilon_{j,i,l} = \mu_j^{(k)} \tilde{r}_{j,i,l} - \tilde{p}_{j,i,l} \quad \text{for} \quad 0 \leq k \leq N-1 \tag{5.18}$$

where

$$\tilde{p}_{j,i,l} = \left(\frac{\mu_j^{(k)}}{\ln 2} - \frac{\Gamma N_0}{s_{j,i,l}^2} \right)^+ \tag{5.19}$$

$$\tilde{r}_{j,i,l} = \log_2 \left(1 + \frac{\tilde{p}_{j,i,l} s_{j,i,l}^2}{\Gamma N_0} \right) \tag{5.20}$$

where $\mu_j^{(k)}$ denotes the Lagrangian multiplier of user j at the kth iteration(level), it will be re-computed after each iteration since the subcarriers sever the jth user possibly change and the received eigenchannel quality will be faded due to block diagonalization for interference-free communication. In other words, at each iteration we assign one subcarrier, limit the number of users to be less than the maximal number of users, and modify the cochannel users' channel quality by

$$\nu_i = \sum_{j=1}^{K} \sum_{l=1}^{r} \epsilon_{j,i,l} \tag{5.21}$$

where ν_i denotes the over all efficiency value of the ith subcarrier and r denotes eigenchannel number over each subcarrier for all user. The larger ν_i is, the more efficiency the ith subcarrier is for all users. Hence, the subcarrier with larger ν need to be decide its assignment such that the deleted users at next iteration can enlarge efficiency value of other subcarriers whose user deletion is not yet operated and which can be arbitrary occupied by all users. Once the subcarrier number is obtained at each iteration, the remaining work is to decide which users will occupy this subcarrier.

For the SISO OFDMA system, just only one user can use a subcarrier at each iteration. Hence, we pick up the user with highest efficiency for this subcarrier which has been discussed in Chapter 4. Now, we focus on MIMO system and must pick up some users under the maximal user number constraint.

We now focus the user selection issue and present some efficient schemes .

5.4.1 Correlation Based User Selection Method via Dual Decomposition

First, we show a simply user selection without considering the orthogonality of users' channels in space domain. In addition, we choose the number of users at each subcarrier to meet the maximal number of users $\hat{K} = N_T/n_r$. Hence, the $\hat{K}$ users whose efficiency is larger than other $K - \hat{K}$ users will be selected at each iteration. After each decision is completed, the channel GNR of the selected users will be modified based on block diagonalization approach and the unselected users will cant't occupy this subcarrier assigned at this iteration. Then, run K OMPA algorithm again and go into next iteration.

However, such user selection scheme don't take the orthogonality into account. Although the selected $\hat{K}$ users have larger efficiency computed by dual decomposition under perfect interference free communication. However, it is possible the channel of these $\hat{K}$ users may be high-correlated, this case will make the channel quality decrease much through block diagonalization in order to fit the interference free communication criteria. So, we propose another user selection scheme which jointly consider the users' channel correlation and the users' channel efficiency.

We propose a second user selection scheme based on the semi-orthogonal user selection (SUS) and all users' spectrum efficiency within each iteration.

Specifically, the transmitter selects the first user from initial user set $\mathcal{A}_0 = \{1, 2, \cdots, K\}$ over the mth subcarrier as

$$\nu_{k,m} = \sum_{l=1}^{\eta} \epsilon_{k,m,l} \quad (5.22)$$

$$\pi(1) = \arg\max_{k \in \mathcal{A}_0} \nu_{k,m}. \quad (5.23)$$

If the maximal number of users can simultaneously over the same subcarrier is equal to one for SISO system discussed in chapter 3, the user selection is completed now.

For MIMO system there are more than one user can transmit over the same subcarrier. Hence, after select p users, if $i < \hat{K}$, the $p+1$th user is selected within the user set

$$\mathcal{A}_p = \{1 \le k \le K : g(h_k, h_{\pi(j)}) \le \mathcal{T}, 1 \le j \le p\} \tag{5.24}$$

as

$$\pi(p+1) = \arg\max_{k \in \mathcal{A}_p} \nu_{k,m}. \tag{5.25}$$

where $\mathcal{T}$ is a design parameter that indicates the maximum spatial correlation allowed between users' channels. In this way, the transmitter will choose users that have high efficiency over current discussed subcarrier and mutually semi-orthogonal. $g(x, y)$ denotes spatial correlation value. For the case x, y are vector in MISO system, $g(x, y)$ can be given by

$$g(x, y) = \frac{|xy^H|}{|x||y|} \tag{5.26}$$

For MIMO system, x, y are matrices, $g(x, y)$ is computed by

$$g(x, y) = \frac{|\hat{x}^H \hat{y}|}{|\hat{x}||\hat{y}|} \tag{5.27}$$

where the columns of $\hat{x}$ represent the right singular vectors of x. We summarize the proposed joint subcarrier (channel) and user assignment algorithm below.

5.5 Numerical Results and Discussion

Some simulated performance of the proposed resource allocation algorithms are presented in this section. We assume the channel matrix with i.i.d zero-mean, unit-variance complex Gaussian entries. For simplicity, we assume that all users are with the same required data rate and BER, i.e. $R_j = R, \forall j$. The same required data rate is possible decided after some scheduling in MAC layer. In Fig. 5.2, we compare the fixed subcarrier assigned and our proposed algorithms in MISO communication with base station has 4 antennas and all mobile terminals are equipped with single antenna. The proposed

Step 1:	(Initial channel allocation) Set $C(j) = \{i \mid 1 \le i \le N\}$, for $1 \le j \le K$. set $S = \{1, 2, \cdots, N\}$ and $t = 0$

Step 2:	(channel selection)
	each user individually invokes the OMPA algorithm based on its SCS $C(j)$
	if $t < N$
	  $i* = \arg_{i \in S} \nu_i$ according to (5.21)
	  Set $S = S \backslash \{i^*\}$, $t = t + 1$, then goto Step 3.
	else goto Step 6.
	end

Step 3:	(initial user election)decide the first user in the i^* subcarrier Set $A_0 = \{1, 2, \cdots, K\}$ and $\nu_{k,i^*} = \sum_{l=1}^{\eta} \epsilon_{k,i^*,l}$ $\pi(1) = \arg\max_{k \in \mathcal{A}_0} \nu_{k,i^*}$. Set $p = 1$ goto Step 4

Step 4:	(user election)
	Set $\mathcal{A}_p = \{1 \le k \le K : g(h_k, h_{\pi(j)}) \le \mathcal{T}, 1 \le j \le p\}$
	if $A_p = \emptyset$ **then** $t = t + 1$ **goto** Step 5.
	else if $p = \hat{K}$ **then** $t = t + 1$ **goto** Step 5.
	else
	  $\pi(p + 1) = \arg\max_{k \in \mathcal{A}_p} \nu_{k,m}$
	  Set $p = p + 1$ and goto Step 4.
	end

Step 5:	(modify GNR via block diagonalization and channel set $C(j)$ for $1 \le j \le K$.) **for** $j = 1 : K$ **if** $j \notin \{\pi\}$ **then** $C(j) \backslash \{i^*\}$ **end** **end** modify GNR via block diagonaization for selected users' indexes $\{\pi\}$ **goto** Step 2
Step 6:	(Output) The final channel allocation and user election

Table 5.1: A joint channel assignment and user selection algorithm

algorithm 2 has be described in Table 5.1. In addition, we show the performance of the proposed algorithm1 in order to see what importance the user selection. The algorithm1 is done by always selecting the $\hat{K}$ users which have better efficient values, where$\hat{K}$ is equal to $N_T/n_r = 4$. The proposed algorithm 2 with user selection is superior to the algorithm 1 without user selection. The reason is that even if the number of eigen-channels in the same subcarrier in algorithm 1 than in algorithm 2, but the users which have good efficient values may be correlated each other. It produces that after block diagonlization the original good channel condition of these users will decrease very much. Simultaneously, we can find the difference among these three schemes increase when the required rate increase. It presents that subcarrier assignment play an important role when the traffic load of system increases and the wireless resource is still limited.

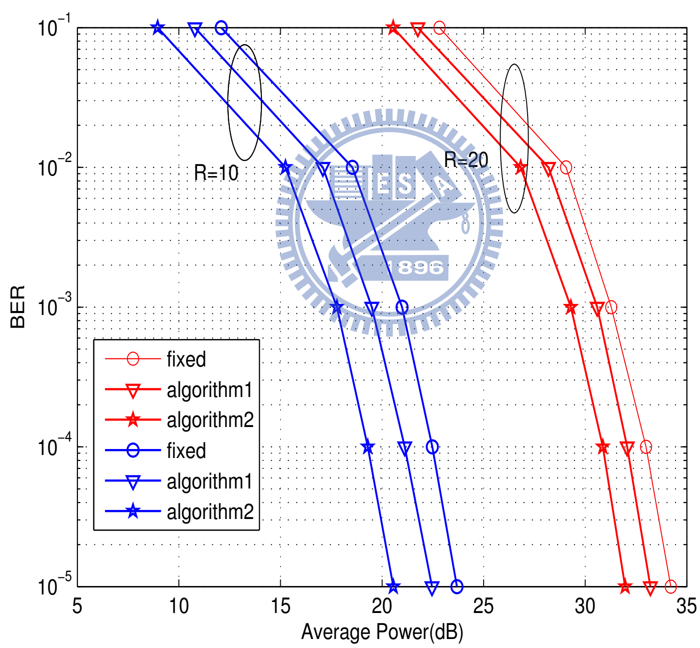

Figure 5.2: Average power per user for the case of 8 subcarriers, 8 users; GNR=0dB, $N_T = 4, n_r = 1$, and $R = 10, 20$ respectively.

Fig. 5.4 plots the effect of the transmit antenna numbers on the system performance. Several performance trends are observed. First, the more the transmit antennas the more important the user selection from the comparison between algorithm 1 and algorithm 2. The difference between fixed assignment and the algorithm 1 decreases when transmit antenna number increasing. In other words, the traditional fixed assignment can improve its poor performance by adding antennas. Finally, this figure shows that the proposed algorithm 2 at the $N_T = 8$ case outperform the traditional fixed assignment at $N_T = 16$ and is close to the performance of the algorithm 2 at $N_T = 16$. For the same performance constraint, we can reduce the transmit antennas by the joint user selection and subcarrier assignment proposed algorithm described in Table 5.1.

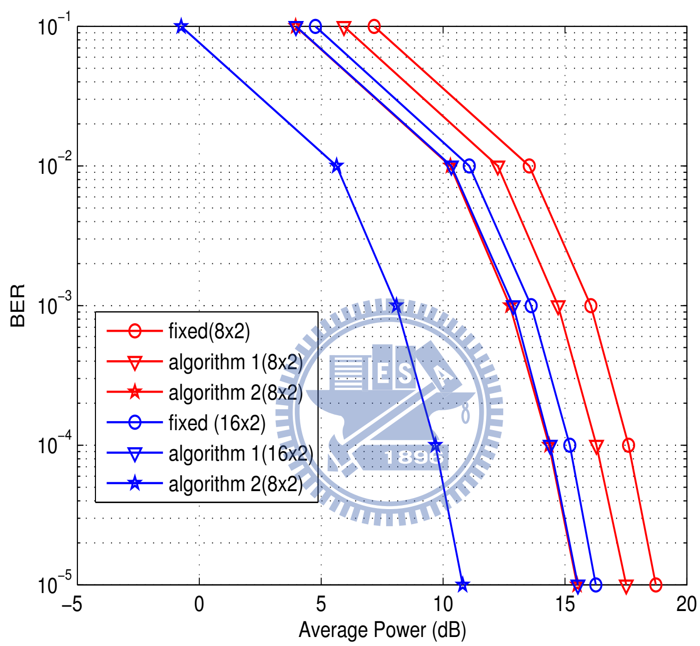

Figure 5.3: Average power per user for $K = 32$, $N = 64$, $R = 20$, $n_r = 2$, GNR=0dB, and $N_T = 8, 16$ respectively.

Next, we discuss the case GNR=-10dB and the other parameters are the same in Fig. 5.4. We can observe that the more transmit antennas are robust to the poor channel conditions. In the same way, the user selection still play an important role in resource allocation.

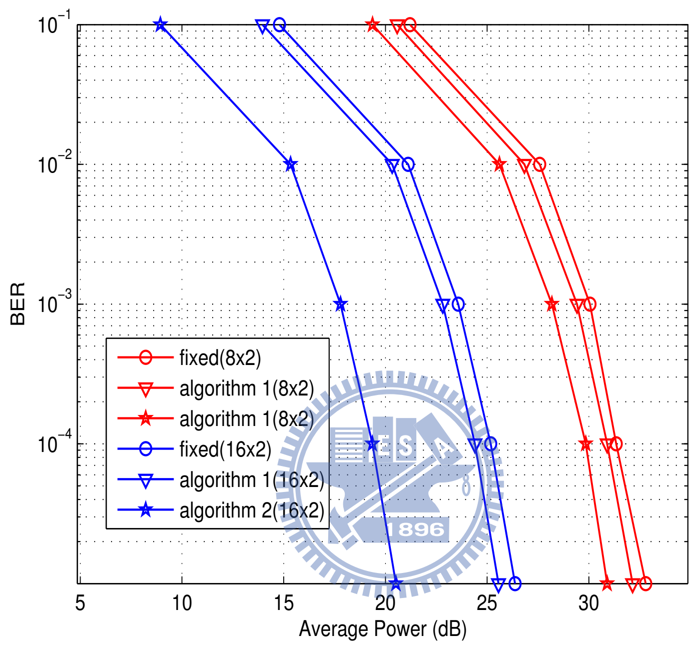

Figure 5.4: Average power per user for $K = 32,\ N = 64,\ R = 20,\ n_r = 2$, GNR=-10dB, and $N_T = 8, 16$ respectively.

Chapter 6

Conclusion

OFDMA is an effective multiple access scheme for wideband wireless mobile networks. Besides its anti-fading capability, an OFDMA system can achieve high spectral efficiency in a multiuser environment by taking advantage of the multiuser diversity and adaptively allocating subcarriers and time slots to the the most suitable users with the minimum required transmit power. Alternatively, one can also maximize the total (weighted) rate or product rate, if fairness is of concern, with power and/or some QoS constraints. Efficient dynamic RA algorithms to solve the above constrained optimization problems in real time is thus crucial for realizing this potential advantage.

Based on the principles of dynamic programming and branch-and-bound, we propose two algorithms–the DPRA and BBRA algorithms–which give either near-optimal or optimal solution. In contrast to the existing algorithms, which suffer from the shortcomings of requiring high complexity and/or unsatisfactory performance, the DPRA algorithm renders near-optimal performance with relative low complexity. Since the existing efficient algorithms are designed with a discrete-rate constraint and use some suboptimal water-filling solution, we make some modifications for fair comparisons. As expected, the resulting ACG and RCG based DPRA algorithms are shown to provide less satisfactory performance with reduced complexities. With proper reuse of the water-filling solution obtained in earlier stages, the average DPRA complexity can be further reduced and is insensitive to d, N and the required sum rate. The average complexity of the

BBRA algorithm, on the other hand, is at least an order higher than that of the DPRA algorithm when the number of users is greater than 10 but is still much less than the known algorithms for obtaining the optimal solution.

The DP-based weighted sum rate or product rate maximization algorithms are as efficient as their counterpart for power minimization. The product rate maximization enjoys the benefit of being highly fair. We also apply the dual decomposition approach to develop low-complexity algorithms for solving the weighted sum rate and product rate maximization problems. Channel ordering is not needed for these cases and the number of calls to single user optimization process (water-filling or OMPA) is at most equal to the product of the subcarrier and user numbers. Finally, we extend our work to MIMO system and propose a low-complexity high performance suboptimal algorithm for MIMO-OFDMA networks.

Our numerical experiment in both independent and correlated fading environments have demonstrated that the near-optimal power-minimization DPRA algorithm is suitable for real-time resource allocation application. In fact, when operating in a more practical correlated fading environment, the outstanding features of the proposed suboptimal algorithms become even more obvious: the performance gains of our algorithms are much higher than those in the ideal i.i.d. fading environments.

On the other hand, the optimal BBRA algorithm is practical only if the user number is small, e.g., $d \leq 5$. Nevertheless, the latter algorithm offers the optimal solution and performance for large N and d with reasonable complexity, which has never been achieved before and is needed for benchmarking and comparison purposes.

Appendix A

An optimal mono-rate power allocation algorithm

Let us redefine the normalized channel capacity r_i by

$$r_i = \log_2\left(1 + \frac{|h_i|^2 p_i}{\sigma_i^2}\right) \equiv \log_2(1 + p_i a_i) \tag{A.1}$$

where the subscript i denotes the ith channel and $|h_i|^2, p_i, \sigma_i^2$ are the corresponding channel gain, transmitted power, and noise power, respectively. In addition, the N orthogonal channels are sorted according to descending channel gain-to-noise ratio, e.g., $a_1 > a_2 > \cdots, a_N$, $a_i \equiv |h_i|^2/\sigma_i^2$. Note that because of (A.1), power and rate allocations are equivalent provided that a_i is known.

For the mono-rate case, (4) becomes

$$\min_{\mathbf{P}} \sum_{i=1}^{N} p_i \ , \quad \text{s.t.} \quad \sum_{i=1}^{N} r_i \geq R, \ p_i \geq 0, \tag{A.2}$$

The water-filling solution implies that only the strong channels (those whose reciprocal channel gains are below the water-filling level) will be used. Hence we assume that only the strongest x channels are used so that the power and rate for the weakest $N-x$ channels are identically zero, i.e., $p_i(x) = r_i(x) = 0$, $x < i \leq N$, where $p_i(x), r_i(x)$ denote the power and rate of the ith channel when only the first x channels are activated. The optimization problem (A.2) then become that determining the optimal x. Define the

Lagrange dual function as

$$f(\{r_i(x)\},\{p_i(x)\},\lambda) = \left[\sum_{i=1}^{N} p_i(x)\right] - \lambda\left[\sum_{i=1}^{N} r_i(x) - R\right] \tag{A.3}$$

and omit the constraints $0 \le p_i$ for the moment. Taking derivative with respect to r_i for $i = 1, 2, \cdots, x$ we obtain $\lambda = \frac{e^{R/x}\ln 2}{\hat{a}(x)}$, where $\hat{a}(x) = \left(\prod_{j=1}^{x} a_j\right)^{1/x}$ and

$$r_i(x) = \frac{R}{x} + \log_2\left(\frac{a_i}{\hat{a}(x)}\right),\ i = 1, 2, \cdots, x \tag{A.4}$$

Obviously, it is possible $r_i(x) < 0$ as the constraint $p_i \ge 0$ has been removed.

Note that

$$r_x(x) = \frac{x-1}{x}\left[r_{x-1}(x-1) + \log_2\left(\frac{a_x}{a_{x-1}}\right)\right] \tag{A.5}$$

Using the fact that, $a_1 \ge a_2 \ge \cdots \ge a_N$, we conclude that

Lemma 1. *The sequence $\{r_x(x), x = 1, 2, \cdots, x\}$ is monotonically decreasing.*

To find the constrained solution we need the following definition.

Definition 4. *An unconstrained solution $\mathbf{r}(x, N) = (r_1(x), r_2(x), .., r_x(x), 0_{1\times(N-x)})$ is said to be admissible if the least rate $r_x(x) > 0$. The admissible active channel number sets for the problem defined by (A.2) is defined by $\mathbf{F} = \{x | r_x(x) > 0, 1 \le x \le N\}$, where $r_x(x)$ is given by (A.4).*

Lemma 2. *The total transmitted power associated with the admissible unconstrained optimal rate assignment (A.4) is a decreasing function of the number of channels used. In other words, $N_1 < N_2 \Longrightarrow \sum_{i=1}^{N_1} p_i(N_1) > \sum_{i=1}^{N_2} p_i(N_2)$, for $N_1, N_2 \in \mathbf{F}$.*

Proof. To begin with, let us assume that $N_1 = m$ and $N_2 = N_1 + 1 = m + 1$, i.e., $N_2 - N_1 = 1$. If the Lemma is valid in this case, it will also be valid when $N_2 - N_1 > 1$.

$$p_i(x) = \frac{e^{r_i(x)} - 1}{a_i},\ 1 \le i \le x, \tag{A.6}$$

The minimum required power for the case $x = m$ is given by

$$
\begin{aligned}
\tilde{P}_m &= \sum_{i=1}^{m} \frac{e^{r_i(m)} - 1}{a_i} = \sum_{i=1}^{m} \left[\frac{e^{R/m}}{\hat{a}(m)} - \frac{1}{a_i} \right] \\
&= m \cdot \frac{e^{R/m}}{\hat{a}(m)} - \sum_{i=1}^{m} \frac{1}{a_i} \qquad \text{(A.7)}
\end{aligned}
$$

where $\hat{a}(m) = [\prod_{i=1}^{m} a_i]^{\frac{1}{m}}$. The minimum required power for the case $x = m+1$ can be expressed as a function of r_{m+1}.

$$
\begin{aligned}
\tilde{P}_{m+1} &= \tilde{P}'_m + p_{m+1} = \sum_{i=1}^{m} \frac{e^{r'i(m)} - 1}{a_i} + p_{m+1} \\
&= \sum_{i=1}^{m} \left[\frac{e^{(R - r_{m+1}(m+1))/m}}{\hat{a}(m)} - \frac{1}{a_i} \right] + p_{m+1} \\
&= m \cdot \frac{e^{(R - r_{m+1}(m+1))/m}}{\hat{a}(m)} \\
&- \sum_{i=1}^{m} \frac{1}{a_i} + \frac{e^{r_{m+1}(m+1)} - 1}{a_{m+1}} \qquad \text{(A.8)}
\end{aligned}
$$

Expressing the difference between $\tilde{P}_m$ and $\tilde{P}_{m+1}$ as a function of $r_{m+1}(m+1)$, we obtain

$$
\begin{aligned}
g(r_{m+1}(m+1)) &= \tilde{P}_m - \tilde{P}_{m+1} \\
&= \frac{m}{\hat{a}(m)} \left(e^{R/m} - e^{(R - r_{m+1}(m+1))/m} \right) \\
&- \frac{e^{r_{m+1}(m+1)} - 1}{a_{m+1}} \qquad \text{(A.9)} \\
g'(r_{m+1}(m+1)) &= \frac{\partial g(r_{m+1}(m+1))}{\partial r_{m+1}(m+1)} \\
&= \frac{e^{(R - r_{m+1}(m+1))/m}}{\hat{a}(m)} \\
&- \frac{e^{r_{m+1}(m+1)}}{a_{m+1}} \qquad \text{(A.10)}
\end{aligned}
$$

The solution of $g'(r_{m+1}(m+1)) = 0$, $r^*_{m+1}(m+1)$, is given by

$$
\begin{aligned}
r^*_{m+1}(m+1) &= \frac{R}{m+1} + \frac{m}{m+1} \cdot \ln \left[\frac{a_{m+1}}{\hat{a}(m)} \right] \\
&= \frac{R}{m+1} + \ln \left[\frac{a_{m+1}}{\hat{a}(m+1)} \right] \qquad \text{(A.11)}
\end{aligned}
$$

For $0 \le r_{m+1}(m+1) < R$, the second derivative of $g(r_{m+1}(m+1))$

$$g^{(2)}(r_{m+1}(m+1)) = \frac{-1}{\hat{a}(m)a_{m+1}} \left[\frac{a_{m+1}}{m}e^{(R-r_{m+1}(m+1))/m} + \hat{a}(m)e^{r_{m+1}(m+1)}\right] \quad \text{(A.12)}$$

is always negative. Since $g'(r^*_{m+1}(m+1)) = 0$, $g'(r_{m+1}(m+1)) > 0$, for $0 \le r_{m+1}(m+1) < r^*_{m+1}(m+1)$, the fact that $g(0) = 0$ then lead to the desired conclusion that $g(r^*_{m+1}(m+1)) > 0$. In other words, the minimum power for the case $x = m$ is larger than that for the case $x = m+1$ which can be achieved with $r_{m+1}(m+1) = r^*_{m+1}(m+1)$. $\square$

The above two Lemmas suggest that the solution to the constrained optimization problem (A.2) can be found by repeatedly calculating the unconstrained solution (A.5) for $x = N, N-1, N-2, \cdots$ until the constraints $p_i \ge 0$, $\forall\ 1 \le i \le x$ are satisfied. A similar but less efficient solution was proposed by Fischer and Huber [2] who iteratively recompute (A.5) by excluding all negative-rate channels and setting $x \leftarrow x - l$, where l is the number of negative-rate channels. Such an approach does not necessarily give the optimal solution and the issue of optimality was not addressed in [2]. Instead of sequentially decreasing x with a decrement of 1, we accelerate the process of locating the optimal x through a bisection search so that the optimal power allocation can be found in Table V.

The resulting algorithm will be referred to as the optimal mono-rate power allocation (OMPA) algorithm henceforth. Note the OMPA algorithm can easily be modified to solve the maximum sum-rate problem

$$\max \sum_i r_i, \quad \text{s.t.} \ \sum_i p_i \le P, \quad p_i \ge 0, \quad \text{(A.13)}$$

Table V: An Optimal Mono-rate Power Allocation Algorithm

Step 1:	(Initialization) Given $a_i, 1 \leq i \leq N$,and R, set $upbound = N$, $lowbound = 1$, and $x^* = [(upbound + lowbound)/2]$.
Step 2:	(Update the lowest rate) $r_{x^*}(x^*) = \frac{R}{x*} + \log_2 \left[\frac{a_{x^*}}{\hat{a}(x^*)} \right]$, where $\hat{a}(x*) = \left(\prod_{j=1}^{x^*} a_j \right)^{1/x^*}$, the number of iterations.
Step 3:	If $r_{x^*}(x^*) \geq 0$, $lowbound \leftarrow x^*$, else $upbound \leftarrow x^*$
Step 4:	If $lowbound < upbound - 1$, $x^* \leftarrow [(upbound + lowbound)/2]$, go to Step 2; else $x^* \leftarrow lowbound$, $r_i(x^*) \leftarrow 0$, for $i > x^*$ and compute $r_i(x^*)$, for $1 \leq i < x^*$.

Bibliography

[1] C. Y. Wong, R. S. Cheng, K. B. Letaief and R. D. Murch, "Multiuser OFDM with adaptive subcarrier, bit, and power allocation" *IEEE JSAC.,* vol 17, pp1747-1758, Oct. 1999.

[2] R. Fischer and J. B. Huber, "A new loading algorithm for discrete multitone transmission," in *Proc. IEEE Globecom*, vol. 1, pp. 724-728, Nov. 1996.

[3] J. Campello, "Practical bit loading for DMT," in *Proc. ICC 1999*, Vancouver, Canada, Jun. 6-10, 1999.

[4] K. Seong, M. Mohseni, and J. M. Cioffi, "Optimal resource allocation for OFDMA downlink systems," *in Proc. ISIT 2006,* Seattle, USA, Jul. 9-14, 2006.

[5] D. Kivanc, G. Li, and H. Liu, "Computationally Efficient Bandwidth Allocation and Power Control for OFDMA," *IEEE Wireless Commun.,* vol. 2, pp. 1150-1158, Nov. 2003.

[6] M. Bohge, J. Gross, A. Wolisz, and M. Meyer, "Dynamic resource allocation in OFDM systems: An overview of cross-layer optimization principles and techniques," *IEEE Network,* vol. 21, pp. 53-59, Jan. 2007.

[7] G. Li and H. Liu, "Resource allocation for OFDMA relay networks with fairness constraints," *IEEE J. Slect. Areas Commun.*, vol. 24, no. 11, pp. 2061-2069, Nov. 2006.

[8] Z. Shen, J. G. Andrews, and B. L. Evans, "Adaptive resource allocation in multiuser OFDM systems with proportional rate constraints," *IEEE Trans. Wireless Comm.,* vol. 4, pp. 2726-2737, Nov. 2005.

[9] M. Ergen, S. Coleri, and P. Varaiya, "QoS aware adaptive resource allocation techniques for fair scheduling in OFDMA based broadband wireless access systems," *IEEE Trans. Broadcasting*, vo l. 49, pp. 362-370, Dec. 2003.

[10] G. Li and H. Liu, "Downlink dynamic resource allocation for multi-cell OFDMA system," *IEEE Trans. Wireless Comm.,* vol. 5, no. 12, pp. 3451-3459, Dec. 2006.

[11] Wimax Forum, "WiMAX Forum mobile relese 1.0 channel model," 2008. (http://www.wimaxforum.org/technology/documents)

[12] C. Y. Ng, and C. W. Sung, "Low complexity subcarrier and power allocation for utility maximization in uplink OFDMA systems," *IEEE Trans. Wireless Comm.,* vol. 7, no. 5, pp. 1667-1675, May 2008.

[13] Z. Mao and X. Wang, "Efficient optimal and suboptimal radio resource allocation in OFDMA system," *IEEE Trans. Wireless Comm.,* vol. 7, pp. 440-445, Feb. 2008.

[14] I. C. Wong, and B. L. Evans,"Optimal Downlink OFDMA Resource Allocation With Linear Complexity to Maximize Ergodic Rates," *IEEE Trans. Wireless Comm.,* vol.7, pp. 962-971, Mar. 2008.

[15] G. Owen,"*Game Theory.*, 3rd ed. New York: Academic Press, 2001.

[16] Z. Han, Z. Ji, and K. J. Ray Liu,"Fair Multiuser Channel Allocation for OFDMA Networks Using Nash Bargaining Solutions and Coalitions," *IEEE Trans. Wireless Comm.,* vol.7, pp. 962-971, Mar. 2008.

[17] Z. Han, Z. Ji, and K. J. Ray Liu,"Dynamic distributed rate control for wireless networks by optimal cartel maintenance strategy," in *Proc. IEEE Globecom,* vol.6, pp. 3454-3458, 2004.

[18] M. Costa,"Writing on dirty paper," *IEEE Trans. Inform. Theory,* vol. 29, no. 3, pp. 439V441, May 1983.

[19] Q. H. Spencer, A. L. Swindlehurst, and M. Haardt,"Zero forcing methods for downlink spatial multiplexing in multiuser MIMO channels," *IEEE Trans. Signal Processing,* vol. 52, no. 2, pp. 461V471, Feb 2004.

[20] L. Choi and R. D. Murch, A transmit preprocessing technique for multiuser MIMO systems using a decomposition approach," *IEEE Trans. Wireless Commun,* vol. 3, no. 1, pp. 20V24, Jan. 2004.

[21] T. Yoo and A. Goldsmith, On the optimality of multiantenna broadcast scheduling using zero-forcing beamforming, *IEEE J. Select. Areas Commun.,* vol. 24, no. 3, pp. 528V541, March 2006.

[22] R. Bellman, *Dynamic Programming.* Princeton, N.J: Princeton University Press, 1957.