AMERICAN UNIVERSITY OF BEIRUT

ORDINALY OPTIMIZED EVOLUTIONARY SCHEDULER

by

NADER RABIH EL-ZARIF

A thesis
submitted in partial fulfillment of the requirements
for the degree of Master of Engineering
to the Department of Electrical and Computer Engineering
of the Faculty of Engineering and Architecture
at the American University of Beirut

Beirut, Lebanon
October 2011

AMERICAN UNIVERSITY OF BEIRUT

ORDINALY OPTIMIZED EVOLUTIONARY SCHEDULER

by

NADER RABIH EL-ZARIF

Approved by:

Dr. Mariette Awad, Assistant Professor Electrical and Computer Engineering	Advisor
Dr. Rabih Jabr, Associate Professor Electrical and Computer Engineering	Member of Committee
Dr. Zaher Dawy, Associate Professor Electrical and Computer Engineering	Member of Committee

Date of thesis defense: October 27^{th}, 2011

AMERICAN UNIVERSITY OF BEIRUT

ORDINALY OPTIMIZED EVOLUTIONARY SCHEDULER

I, NADER RABIH EL-ZARIF

☐ authorize the American University of Beirut to supply copies of my thesis to libraries or individuals upon request.

☐ do not authorize the American University of Beirut to supply copies of my thesis to libraries or individuals for a period of two years starting with the date of the thesis defense.

Signature

Date

Acknowledgments

First, I would like to express my deep gratitude to Prof. Mariette Awad for supervising me and helping me during this thesis and for her assistance ever since I joined the American University of Beirut masters program. Second, I would also like to express my appreciation to Prof. Zaher Dawy and to the PHD candidate, Ahmad El-Hajj for accompanying me, step-by-step, during all the thesis work.

I would like to extend my thanks to committee member, Prof. Rabih Jabr who gave me constructive remarks during the proposal that helped me achieve a better work.

Last but not least, I would like to thank my friends who supported me all the time especially during periods of difficulty and to my parents who provided me with all the support and love since I was a child.

An Abstract of the Thesis of

Nader Rabih El-Zarif for Master of Engineering

Major: Electrical and Computer Engineering

Title: Ordinally Optimized Evolutionary Scheduler

With the emergence of OFDMA-based wireless standards such as the 3GPP Long Term Evolution (LTE) and the IEEE 802.16 (WiMAX), the number of users requesting higher data rates and lower delays has drastically increased. The satisfaction of such Quality of Service (QoS) requirements mandates an efficient allocation of the resources among active users.

In particular, this thesis addresses the problem of downlink resource allocation in OFDMA-based networks. While most existing work relied on mathematical optimization and game theory, this work uses a novel combination of ordinal optimization (OO) and the well-known genetic algorithm (GA) to develop efficient downlink resource allocation schemes. OO technique is used to select the initial GA population and stopping criteria for more reliable convergence. The number of possible carrier allocation is al-

most infinite; to sample from such a large space, OO is modified to handle more than 1000 samples. The formulated GA problem proposes a new fitness function that aims at maximizing throughput while providing priorities for real-time users and considering previous allocation outcomes to ensure long-term fairness among users. GA parameters such as the initial population and the stopping criteria are determined using OO techniques.

The comparison between the proposed “Ordinaly Optimized” evolutionary scheduler and existing work shows improved fairness while maintaining the overall throughput at an acceptable level for different simulated channel conditions.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

The need for bandwidth is increasing exponentially. According to Cisco forecasting, mobile data traffic is doubling year to year through 2014, and between 2009 and 2014 it would have increased by 39 times. Today, smartphones generate as much traffic as 24 basic-feature phones in view of the introduction of mobile services, such as video streaming, video call, mobile TV, and video conferencing [1]. Thus a network optimized for mobile voice only, won't be able to efficiently handle high numbers of mobile internet users without high speed data communication capabilities. Of course wired access such as xDSL and fiber optics can achieve data rate of several gigabits per second without suffering from radio signal attenuation, fading, scattering, reflection, interference as wireless access does. However the high cost to pay is the lack of mobility, which in modern life is not easily accepted.

Orthogonal frequency division multiplexing (OFDM) is a promising solution that provides high speed wireless connection. By dividing the original stream into several parallel streams, OFDM can combat inter-symbol interference (ISI) and frequency selective fading. OFDM can prove very useful when combined with multiple input multiple output (MIMO) technology. Note that OFDM is a modulation scheme that can only support single user for a specific symbol duration; on the other hand, orthogonal frequency division multiple access (OFDMA) is used to make OFDM a multiple access scheme that supports mobility. The OFDM/OFDMA technology presents a significant advantage to wireless media such as WiMAX, WiFi, LTE (downlink) and many more. However, in spite of the advantage of OFDM and OFDMA, their major limitation is that their waveforms have high peak-to-average power ratio.

In wireless access radio resources are scarce and the throughput is upper bounded by Shannon limit. It is becoming nearly impossible to meet all users demands as the penetration rate among population increases. The area where optimization plays a key role is in data scheduling. In a typical wireless system, there are multiple mobile stations (MSs) that request services with different quality of service (QoS) requirements. The base station (BS) needs to have a scheduling policy that takes into consideration the different QoS requirements such as data rate, latency and error rate. The packets could also result from different types of applications: video streaming, voice over IP (VOIP), web browsing, emails and file transfer protocol (FTP). The QoS for these applications are quite different. The scheduler needs to allocate these packets differently so their QoS requirements can be met while taking into account the power and throughput constraints. Because wired scheduling algorithms are not efficient in wireless medium, as they don't take into account channel characteristics, the wireless standard is left open for novel resource allocation and scheduling approaches, which makes scheduling and resource allocation a hot research topic.

In this thesis we propose investigating a scheduling and resource allocation algorithm for a congested downlink OFDMA scenario, where every user has different service requirements and different channel conditions. We propose combining OO & GA into an "ordinaly optimized" evolutionary scheduler (OOES) to serve two classes of users: the real-time and the non-real-time users. We presume that during each frame period, the base station BS will receive the requests and the channel state information (CSI) from the users; and since the radio resources are scarce, the BS will perform scheduling and allocate resources to the MS by satisfying the requested QoS while maximizing fairness in service.

In the published work, most of the literature on scheduling in OFDMA system that used GA assumed that we have real time and non-real time users without differentiating users' priority within the same service class. In addition, most of the publications didn't compensate for under-served users. To solve these problems, we introduce a fitness function to incorporate two parameters: "time to service" and "residue". The "time to service" parameter stores the "maximum remaining time" for each real time user to get serviced before degradation on the requested service occurs. Thereby, some real time users will have more priority over others depending on the service used. The "residue" parameter stores the difference between the requested throughput and the obtained data rate for each user, in order to compensate the under-allocation as soon as possible. The main problem in carrier allocation in OFDMA systems is that the search space of the system is exponential with the number of users or the number of carriers, and the number of "good" carrier allocation is very narrow. A combination of OO and GA is used to help in achieving better carrier allocation. Note that OO was never used before in scheduling in OFDMA systems. The combined OO-GA solution was used by [2] to solve a flow shop scheduling problem. However OOES uses OO to determine the initial population of GA in addition to the stopping criteria instead of just the stopping criteria.

The following chapters describe our proposed approach to the subject matter at hand. The literature review and background concepts on OO, GA, OFDM and scheduling in wireless systems are presented in chapter 2. The problem formulation and work-flow are tackled in chapter 3. Analysis of the proposed algorithm are presented through simulation in chapter 4. Finally, in chapter 5, research conclusions are emphasized.

Chapter 2

Background Concepts and Literature Review

The following chapter offers background concepts on OO, GA, Hammersley sampling, OFDM and OFDMA. The purpose of the literature review is to offer a review of currently used OO and GA methods and techniques with emphasis on how they managed to improve their performance.

2.1 Background Concepts

2.1.1 Ordinal Optimization

The concept of OO was explained in [3, 4]. Its main idea is that, instead of searching for the best solution, a good enough solution with high probability is sought. Its main advantage is the convergence time while maintaining a predefined level of confidence in the result. Before detailing the OO procedure, let's define the following terminologies:

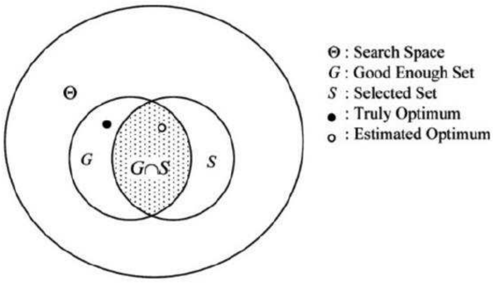

Figure 2.1: OO Basic Principle [5].

- Θ: The search space, which is considered to be a very large finite set.
- N: The number of designs uniformly chosen in Θ.
- G: The good enough set, usually the top-g elements in the design space.
- S: The selected designs in N.
- K: The alignment level.
- AP: The alignment probability $= Prob\left[G \cap S \geq K \,/N, C, \sigma^2\right]$.
- σ^2: Noise level.
- Order performance curve (OPC): Plot of the value of the performance as a function of the order of performance.
- C: Class of OPC for the problem shown in figure 2.2.
- Universal alignment probability (UAP) $=Prob\left[G \cap S \geq K \,/N, C, \sigma^2\right]$
- Selection rule: The method that is used to select the subset S.

The selection rules can be categorized into two main categories: blind pick (BP) and horse race (HR). In BP scenario we have no idea about the actual performance; in other words, the observed performance is assumed to have infinite noise variance. The alignment probability for BP is:

$$AP = \sum_{i=K}^{min(G,S)} \frac{\begin{pmatrix} G \\ S \end{pmatrix} \begin{pmatrix} N-G \\ S-i \end{pmatrix}}{\begin{pmatrix} N \\ S \end{pmatrix}} \tag{2.1}$$

if a given alignment probability, as well as size of the good enough set and alignment level are known, one can randomly pick S designs from N, after determining the size of the selected subsets.

In HR selection rule the noise level is not infinite and hence, the selected subset S is not chosen at random anymore. The procedures of HR selection rule is as follows:

1. Sample N designs from Θ uniformly and randomly.
2. Use a crude model to estimate the performance of these N designs.
3. Specify G and K.
4. Estimate the most appropriate OPC class of the problem shown in figure 2.2 and the noise level of the crude model (low, medium or high).
5. Calculate the size of the selected set according to equation 2.2 and figure 2.3.
6. Select the observed top S designs in N.

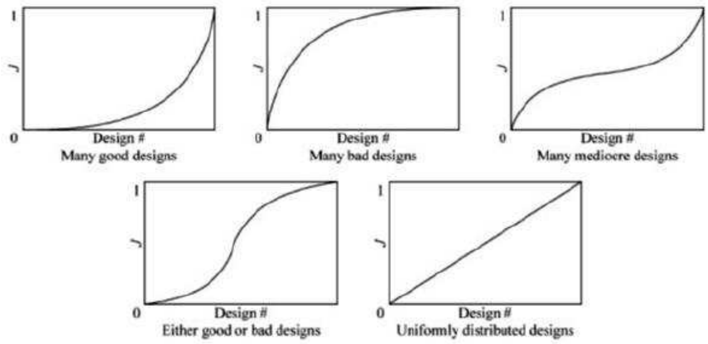

Figure 2.2: Five types of the OPC classes [5].

According to [4], by the use of statistical methods and regression the size of the selected subset is:

$$S = Z(G, K) = e^{Z_1} K^{Z_2} G^{Z_3} + Z_4 \tag{2.2}$$

Z_1, Z_2, Z_3 and Z_4 are constants of regression that depend on the OPC class and noise level. Their values can be found according to figure 2.3. Note that the size of the selected subset in HR rule is upper bounded by the BP [5].

Noise	∞	$U[-0.5,0.5]$				
OPC class	B-Pick	Flat	U-shape	Neutral	Bell	Steep
Z_1	7.8189	8.1378	8.1200	7.9000	8.1998	7.7998
Z_2	0.6877	0.8974	1.0044	1.0144	1.9164	1.5099
Z_3	–0.9550	–1.2058	–1.3695	–1.3995	–2.0250	–2.0719
Z_4	0.00	6.00	9.00	7.00	10.00	10.00
Noise	∞	$U[-1.0,1.0]$				
OPC class	B-Pick	Flat	U-shape	Neutral	Bell	Steep
Z_1	7.8189	8.4299	7.9399	8.0200	8.5988	7.5966
Z_2	0.6877	0.7844	0.8989	0.9554	1.4089	1.9801
Z_3	–0.9550	–1.1795	–1.2358	–1.3167	–1.6789	–1.8884
Z_4	0.00	2.00	7.00	10.00	9.00	10.00
Noise	∞	$U[-2.5,2.5]$				
OPC class	B-Pick	Flat	U-shape	Neutral	Bell	Steep
Z_1	7.8189	8.5200	8.2232	8.4832	8.8697	8.2995
Z_2	0.6877	0.8944	0.9426	1.0207	1.1489	1.3777
Z_3	–0.9550	–1.2286	–1.2677	–1.3761	–1.4734	–1.4986
Z_4	0.00	5.00	6.00	6.00	7.00	8.00

Figure 2.3: Regressed values of Z_1, Z_2, Z_3 and Z_4 [5].

2.1.2 Genetic Algorithm

GA was formally introduced in [6]; it works very well on mixed combinatorial problems. It is less susceptible to getting 'stuck' at local optima than gradient search methods, but tends to be computationally expensive. The main challenges in applying GA to real life problems is selection of the fitness function, the transition from the phenotype to the genotype, and finally the parameter settings for the genetic operators (selection, mutation and crossover). The problem parameters need to be encoded inside each chromosome so that every chromosome represent a possible solution. Then, by applying genetic operators such as mutation and crossover, and with the offspring, the population of chromosomes evolves to eventually converge to

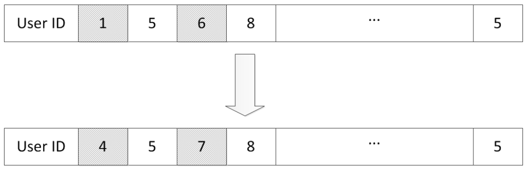

Figure 2.4: Mutation operator

the desired solution. The steps required in GA are the following:

1. Encode the chromosomes: An individual is characterized by a set of parameters: genes. The chromosome contains all information needed to construct the fitness or performance. To construct a chromosome, "feasible" solutions are encoded into genes. Which are joined into a string: encoded chromosome.

2. Select the initial population: Usually random data within the search space is encoded into chromosomes to generate the initial population.

3. Perform mutation: Mutation is when the value of the selected gene is randomly changed as shown in figure 2.4. Each gene has a predefined probability of mutation. Mutation is used to explore possible solutions in different areas of the search space.

4. Perform crossover: In crossover, chunks of the chromosomes are exchanged as shown in figure 2.5. The purpose of crossover is to generate better chromosomes by swapping "bad" genes with "good" ones.

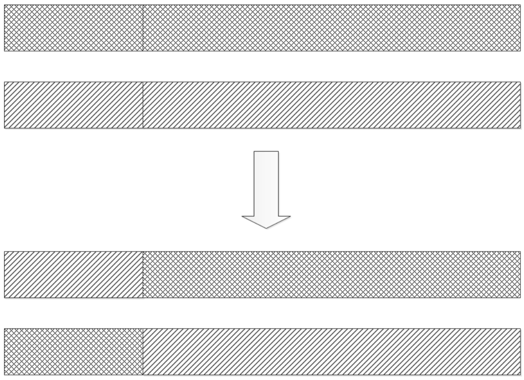

Figure 2.5: Single point crossover

5. Perform chromosome selection: In this stage the selected genes are chosen for the next generation. In roulette wheel selections, the probability of the chromosome to be selected is proportional to its performance as shown in figure 2.6.

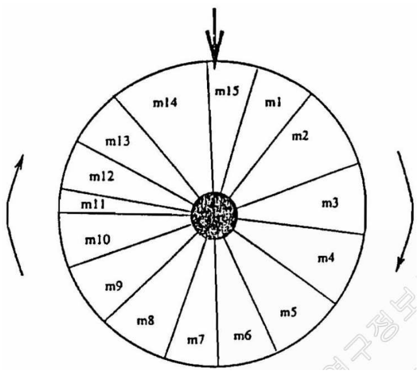

Figure 2.6: Roulette wheel selection

6. Repetition until objective is met: The process is repeated until when of the following criteria is met:

   (a) GA converges to the desired solution

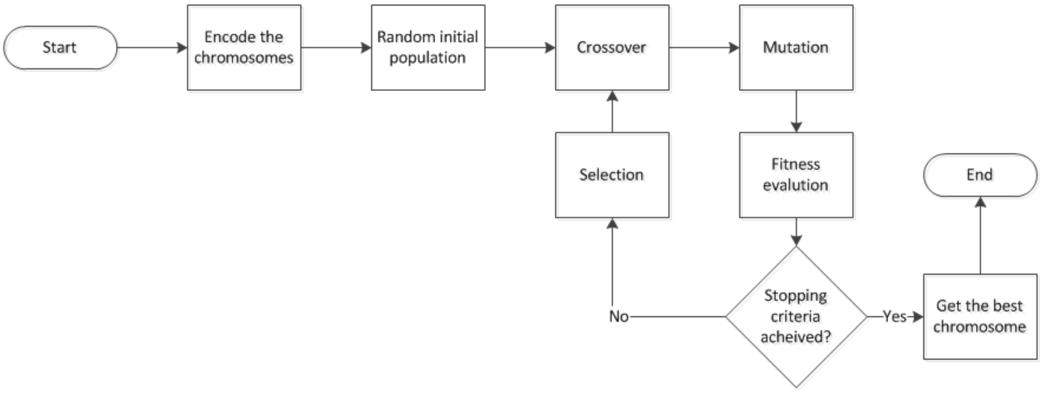

Figure 2.7: GA flowchart

(b) There is no more change in performance for a specified number of generations.

(c) The time limit expires.

(d) The error drops to a certain limit.

These conditions are difficult to estimate in a problem where there is no prior knowledge.

The flowchart of GA is shown in figure 2.7.

2.1.3 Hammersley Sampling

Halton and Hammersley points are useful methods to uniformly sample data points. According to [7] every positive integer k can be expanded using a prime base p:

$$k = a_0 + a_1 p + a_2 p^2 + a_3 p^3 + \cdots + a_r p^r \tag{2.3}$$

where each a_i is an integer in $[0, p-1]$. Function $\Phi_p(k)$ is defined as:

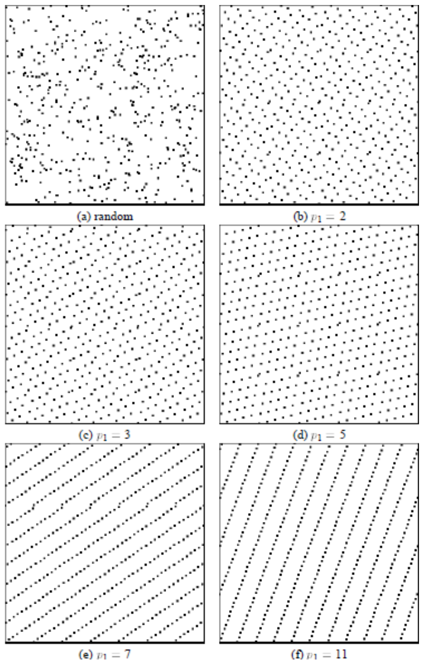

Figure 2.8: Effect of p on Hammersley points [7]

$$\Phi_p(k) = \frac{a_0}{p} + \frac{a_1}{p^2} + \frac{a_2}{p^3} + \cdots + \frac{a_r}{p^{r+1}} \tag{2.4}$$

For a d dimensional data one can have $p_1, p_2, \cdots, p_{d-1}$, then one can compute their corresponding sequence $\Phi_{p_1}(k), \Phi_{p_2}(k), \cdots \Phi_{p_{d-1}}(k)$, finally a set of n Hammersley points is obtained:

$$\left(\frac{k}{n}, \Phi_{p_1}(k), \Phi_{p_2}(k), \cdots \Phi_{p_{d-1}}(k)\right) \qquad for\ k = 0, 1, 2, \cdots n-1 \tag{2.5}$$

The main problem in Hammersley points is that with the increasing value of p, the points will align as shown in figure 2.8.

From figure 2.8, it is noticeable that random points are dense in

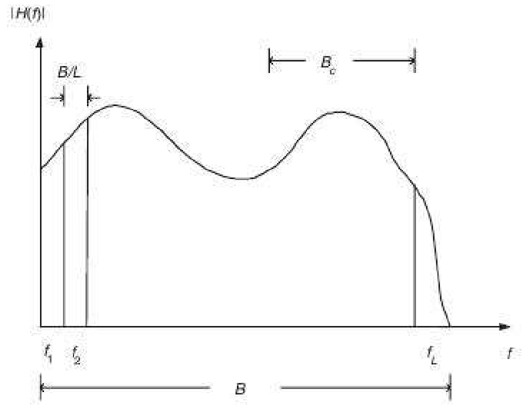

Figure 2.9: Multicarrier modulation [9]

some areas and non-existent in other, while Hammersley points look more uniform with lower variance. The Halton sampling technique is similar to Hammersley's, but the generated points will have higher variance and will not suffer from the alignment problem. The main application of both of these techniques is image reconstruction. In addition, Halton points are also utilized in [8] for direct illumination.

2.1.4 OFDM And OFDMA

The idea of OFDM is based on multicarrier modulation. That is the original stream is divided into several parallel sub-streams with low data rate, instead of transmitting it directly. The number of sub-streams is chosen to ensure that each sub-channel has a bandwidth less than the coherence bandwidth of the channel, so the sub-channels experience relatively flat fading [9] as shown in figure 2.9 .

Figure 2.9 shows that by dividing the original stream into L parallel streams, the channel response looks more flat, which in turn can help

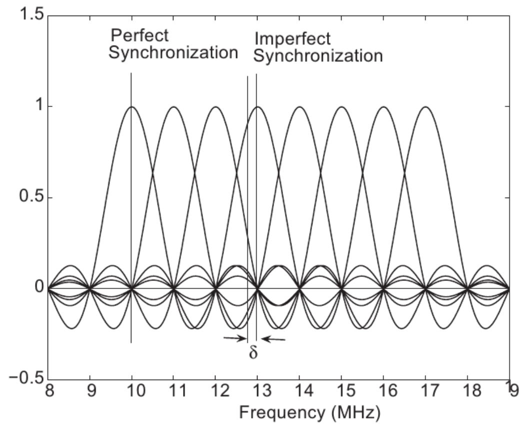

Figure 2.10: Perfect synchronization and synchronization error [9]

to remove the ISI. The additional advantage of OFDM is that the carriers are overlapping instead of being spaced apart, which in turn can save bandwidth. However, the carriers must be orthogonal and perfectly aligned and synchronized in order to prevent them from affecting each other performance as shown in figure 2.10. That is when the amplitude for one of the carriers is maximized while it must be zero for all other carriers.

2.2 Literature Review

2.2.1 Scheduling in OFDM Systems

In the following literature we provide an overview of the previous research that studied scheduling in wireless system. The most relevant work is presented addressing constraints scheduling as proposed by [10]. Authors in [10] reported that schedulers must be consistent with WiMAX system, meet the fairness among users, satisfy the QoS requirement of each service class, maximize system throughput, minimize power consumption, and be as simple as possible. As contradictory and somewhat redundant as these constraints seem, these goals must be met to ensure scalability.

The first issue is how to achieve fairness among users. The authors in [11] proposed a two-stage fair and effective queueing (FEQ) algorithm: the first stage used weighted round robin to provide minimum rate required (MRR) for each user, while the second phase utilized earliest deadline first for the remaining bandwidth to minimize the packet dropping rate. Despite the fact that fairness is a very important issue, the algorithm didn't take CSI into consideration. Indeed, a user with a bad channel condition will consume a larger proportion of the available bandwidth to achieve its MRR, which decreases the throughput of the system.

The second issue is how to mitigate the channel effect while meeting the QoS requirement for real-time polling service (rtPS) users. G. Arul Prasath et al. [12] proposed a solution based on introducing a parameter α that is used to control the bandwidth division between real-time and non-real-time traffic. As the distance between BS and MS increases, the delay increases, and hence the QoS of real-time traffic decreases; as a result, α varies to give higher priority to real-time traffic to increase its QoS without causing congestion for the non-real-time polling service (nrtPS) buffer.

Another approach to throughput maximization and maintaining the QoS was performed in [13]. The authors formulated a scheduling algorithm that consists of combining temporary removal scheduler with modified maximum signal to interference ratio (mmSIR). Their result showed that mmSIR has similar performance as mSIR in terms of overall system throughput and number of served subscriber stations (SS) per frame, and an improvement in terms of mean sojourn time which helps to meet the QoS requirement of rtPS users. To reduce the frame occupation ratio, authors in [14] proposed an algorithm based on channel learning, although it is designed for WiMAX classes, the algorithm didn't take the queueing delay into consideration. To meet the QoS requirements of service classes while maintaining the system throughput, and to increase the overall transmission control protocol throughput while achieving fairness among users, the authors in [15] proposed joint optimization between link adaptation, automatic repeat request for scheduling algorithms.

The issue of ensuring fairness while satisfying the QoS requirement of each service class, was tackled by R. Jayaparvathy et al. [16] from a different point of view. The authors suggested an approach based on dynamic weight adjustment scheduling algorithm derived from Nash equilibrium. Every SS supports four classes of traffic: unsolicited grant service , rtPS, nrtPS and best effort (BE). As each class has its own queue and its own weight, they compete for the bandwidth provided. The weights depend on the QoS specification and congestion. As nrtPS traffic increases, its weight rises, and hence, the QoS of rtPS and BE traffic drops accordingly. Although the link budget is used in the simulation, the modulation order is not taken into consideration. However, this approach fails short of resolving the issue of maximizing throughput and minimizing power consumption.

Other game-theoretical based approaches were introduced by [17, 18, 19] to solve the issue of throughput maximization and improvement of resource allocation among users. The authors proposed game theoretical channel-aware schedulers using different games. In [17] the scheduling is performed in two steps: inter-class and intra-class scheduling. In the former class, the users sort their packets in decreasing order of utility, while in the latter, the scheduler divide the available resources among the users by using the principle of game theory. Their result showed that the Nash solution provides a good fairness among the users with high resource utilization. In [19], [20] Nash bargaining solution is used where the BS enforce cooperation among the users. The BS chooses the users whose rate maximizes the summation of utilities, where the utility function is linear [19] or logarithmic [20] to ensure proportional fairness among users vs. relative throughput maximization. The authors in [19] showed that the Nash bargaining solution (NBS) ensures synchronized fairness and high sum-rate. However, as the number of connections increases, the sum-rate decreases to nearly proximate the max-min fairness solution, and the fairness variance decays to zero. In [20] the results were simulated and it was shown that NBS is just another version of proportional fairness.

A different game theoretical approach in [18] is based on a non-cooperative game which proposes to solve the scheduling problem by correlating call admission and bandwidth allocation among users. This correlation makes sure that the QoS doesn't drop below a certain predefined threshold. For real-time applications, the QoS is the delay, as for non-real-time application the QoS is the throughput.

2.2.2 Scheduling In OFDMA System Using GA

Other authors chose GA to solve the scheduling problem. The authors in [21] performed joint optimization between the scheduling and link adaptation using GA, disregarding the delay and QoS requirements of each service class. Y. Teng et. al [22] proposed improved complexity reduced genetic algorithm (CRGA) to achieve fairness among real-time traffic users and non-real-time traffic users while maximizing the system utility. In CRGA the chromosome data is not binary anymore (a vector is used to note user index for each carrier) which reduces the chromosome size making the crossover operation much simpler. Because of that, CRGA converges much faster than the traditional GA. The algorithm allocates more carriers to the real-time users when the channel condition is bad, and increases the data rate of non-real-time users when the channel is good. However, the algorithm does not compensate for users who receive low data rate during the scheduling period.

The resource allocation objective is to allocate subcarriers, bit and power dynamically according to the instantaneous channel condition for every user. To maximize the system throughput the authors in [23] proposed waterfilling algorithm as fitness function for GA to allocate carriers to the users. A random initial population is created then a good individual is added; this individual will help GA to converge faster and to obtain a better solution. However, the proposed algorithm does not take into account real-time users.

To mitigate the long convergence time of GA, the authors in [24] proposed an algorithm that combines Karush-Kuhn-Tucke (KKT) conditions

[25] with GA to perform resource allocation in wireless mesh networks. The algorithm proposed in [24] simply compares KKT-driven approach with GA after 10 iterations, and then chooses the best one. If GA was favored during the scheduling, it will continue to reach 25000 iterations. Therefore, the performance of combined KKT-GA is lower-bounded by KKT and upper-bounded with GA with relatively low order complexity. Simulations were done to prove this fact.

Another way to achieve fairness among users while maximizing the utility is proposed by the authors in [26] who conceived a joint optimization based on GA. The stochastic approximation was used to control the parameters in scheduling, and GA was utilized to improve the carrier allocation among the users. Simulations showed that the proposed algorithm outperformed M_LWDF in terms of average delay and packet loss rate. Even though the delay was reduced, which makes the algorithm suitable for real-time services, the author didn't show how the data rate is improved. From their part, Mehrjo et al. [27] proposed the use of GA to allocate carriers to real time and non-real time users where the former has a sigmoid-like utility and the latter has a logarithmic utility.

Finally, to meet the QoS requirement of IEEE 802.16 users, Y. Chiu et. al in [28] suggested the use of GA with subscriber station (SS) grouping resource-allocation (GGRA). Since most of 802.16 traffic is delay-sensitive, the authors in [28] used the notation of residual life-time to increase the priority of the users that did not receive allocation during frame period, in the way the QoS of all user classes are met. The algorithm first aggregate highly correlated SSs together by the means of virtual MIMO system, and then assigns each SS from the same group to different slots to prevent high mutual interference and minimize the number of genes in GA, which in turn performs allocation based on the increasing order of residual life-time. The result showed that GGRA outperformed maximum largest weighted delay first (MLWDF), and efficient and fair scheduling (EFS) in terms of system throughput, ratio of unsatisfied hypertext transfer protocol (HTTP) users and FTP throughput. Even though the GGRA run-time was higher than that of EFS, it was able to provide the allocation within the 5-ms frame duration requirement.

2.2.3 Published Fitnesses and Objective Functions

Fitness 1

Authors in [23] proposed the waterfilling algorithm as an objective function, that is to maximize the total transmitted power:

$$Objective = \underset{b_{k,n}}{\operatorname{argmax}} P_{total} = \sum_{n=1}^{N_C} \sum_{k=1}^{K} \left(\frac{f\left(b_{k,n}\right)}{\alpha_{k,n}^2} \right) \tag{2.6}$$

where $b_{k,n}$ is the number of bits allocated to user k using subcarrier n, and $\alpha_{k,n}$ is the channel gain of user k using subcarrier n.

$$f\left(b_{k,n}\right) = \frac{N_0}{3} \left[Q^{-1} \left(\frac{BER_n}{4} \right) \right]^2 \left(2^{b_{k,n}} - 1 \right) \tag{2.7}$$

where BER_n is the bit error rate for subcarrier n, Q^{-1} is the inverse Q function, and N_0 is the noise power spectral density.

Fitness 2

Authors in [27] assumed a model of K number of users of which K' are real time users. Their objective is to maximize the utility as shown below:

$$Objective = max\left(\sum_{k=1}^{k=K} U_k\left(r_k\right)\right) \tag{2.8}$$

Where r_k is the data rate allocated to user k. The utility is given by:

$$U_{real} = \begin{cases} 0 & r \leq l_1 \\ \sin^k\left(\frac{\pi}{2} \times \frac{r-l_1}{l_2-l_1}\right) & l_1 < r \leq l_2 \\ 1 & r > l_2 \end{cases} \tag{2.9}$$

$$U_{non\_real} = \begin{cases} \log\left(1 + 10^{-6r}\right) & r \leq l_3 \\ 1 & r > l_3 \end{cases} \tag{2.10}$$

Where $l_1 = 250Kbps$, $l_2 = 5Mbps$, $l_3 = 9Mbps$.

Fitness 3

Authors in [22] assumed a model of K number of users of which K' are real time users. Their objective is to maximize the utility as shown below:

$$Objective = max\left(\sum_{k=1}^{k=K'} U_k\left(r_k\right) + \lambda \times \sum_{k=K'+1}^{k=K} U_k\left(r_k\right)\right) \tag{2.11}$$

Where λ is a parameter that is used to vary the weight of real-time users, and r_k is the total number of bits allocated to user k. Note that if the overall channel is bad, λ should change to give more weight for real time users. λ is defined by:

$$\lambda = \frac{\sum_{k=1}^{k=K'} U_k\left(r_k\right)}{\sum_{k=K'+1}^{k=K} U_k\left(r_k\right)} \tag{2.12}$$

The utility is given by:

$$U_{real} = \begin{cases} 0 & r \leq 20 \\ \frac{1}{1+e^{\frac{1}{10-0.08r}}} & 20 < r < 200 \\ 1 & r \geq 200 \end{cases} \tag{2.13}$$

$$U_{non\_real} = \begin{cases} 0.5\log_{10}\left(1 + 10^{-2r}\right) & r < 638 \\ 1 & r \geq 638 \end{cases} \tag{2.14}$$

2.2.4 GA and OO

In the following, we state some of the works that tried to improve the performance of GA and OO. First, to improve the performance of GA, the author in [29] developed an adaptive real code genetic algorithm (ARGA). The algorithm first specifies the key parameters in GA such as: crossover probability, mutation probability, selection size, and population size; then it classifies the parameters into important parameters which greatly affect the performance of GA, and unimportant one that have very little effect on GA. Finally, it classifies the important parameters into sensitive (have different effect on GA at different stages) and robust (have similar effect on GA at different stages). By identifying sensitive parameters, the algorithm can modify them throughout the process whenever it is necessary. The performance of ARGA is validated through simulations. Second, to improve the performance of OO, the authors in [30] used iterative OO to solve a stochastic optimization problem. Their idea is to narrow the search into subsets, and then favor the good subsets of the search space through limited sampling. The process is repeated iteratively in order to obtain a much smaller subspace of in which we can find the solution. The algorithm is also applied to the Witsenhausen problem (1968) -which is a multidimensional problem- and managed to find a “good enough" solution for it.

The authors in [31] and [2] proposed an algorithm that combines the features of OO and GA to solve mixed integer programming (MIP) and flow shop scheduling respectively. The MIP and flow shop scheduling are problems that usually belong to non-deterministic polynomial time (NP) class. To solve such problems, OO is used to determine the number of iterations for GA given the required performance of the solution and the confidence level. The number of generations in GA is proportional to the number of iterations provided by OO and inversely proportional to the chromosome size. After determining the size of the initial population and the number of generations, the authors need to determine a suitable mutation operator. In [31] the mutations are done by using triangular distribution since the data is integer not binary; then, in the final stage the MIP problem will be transformed to linear programming (LP) problem which can be easily solved; while, in [2] two stages of mutations are used to improve the obtained result. Simulations were done in production planning and scheduling in batches and in flow shop to validate the proposed models.

Chapter 3

Proposed Workflow

In this chapter, the proposed methods to perform scheduling in OFDMA system are explained. First, problem formulation and the proposed fitness function is presented with its details. Second, the search space is sampled to select the initial chromosome population. Two types of sampling techniques are proposed: direct sampling and two-level sampling. Third, different GA operators such as the initial population, the number of generations, and the mutation operator are modified in order to generate a "feasible" carrier allocation.

3.1 Proposed Fitness and Objective Function

Before we go through problem formulation, we introduce the following terminologies let:

1. K, N and P represent the total number of users, total number of carriers and maximum available transmitted power at the base station

respectively.

2. $p_{k,n,t}$ be the transmit power allocated to user k on subcarrier n and scheduling frame t.

3. $c_{k,n,t}$ be the allocation of user k on subcarrier n and scheduling frame t.

4. $\gamma_{k,n,t}$ be the channel quality of user k on subcarrier n and scheduling frame t.

5. $M_{k,n,t}$ be a parameter that is proportional to the number of bits per subcarrier n for user k at scheduling frame t.

6. $\rho(.)$ be the data rate which can be achieved with the transmit power $p_{k,n,t}$.

7. $H_{k,n,t}$ be the channel gain of subcarrier n if user k is using subcarrier n at scheduling frame t.

8. N be the noise power spectral density.

9. $SNR_{k,n,t}$ be the received signal to noise ratio at user k if it is using carrier n at a scheduling frame t.

10. $N_{k,t}$ be the desired data rate to user k at scheduling frame t.

11. $T_{k,t}$ be the maximum time for the user to get serviced after a scheduling frame t.

12. T_f be the frame duration.

13. $X_{k,t}$ be the maximum number of frames for the user to get serviced after a scheduling frame t.

14. $D_{k,t}$ be the demand of user k at scheduling frame t.

15. $U_{k,t}$ be utility of user k at scheduling frame t.

16. $R_{k,t}$ be the dissatisfaction with the service for user k at a scheduling frame t.

The objective is to find the carrier allocation $c_{k,n,t}$ that maximizes the total sum of difference between the users utility and the dissatisfaction with the service from the previous carrier allocation.

$$\arg\max_{c_{k,n,t}} \sum_{k=1}^{K} (U_{k,t}(c_{k,n,t}) - R_{k,t}) \tag{3.1}$$

subject to the following constraints:

1. For a given frame, a carrier can only be allocated to one user.

$$\sum_{k=1}^{K} c_{k,n,t} = 1 \quad \forall n \in [1, N], \forall t \tag{3.2}$$

2. For all frames, the number of carriers allocated to all users cannot exceed the total number of carriers.

$$\sum_{k=1}^{K} \sum_{n=1}^{N} c_{k,n,t} \leq N \quad \forall t \tag{3.3}$$

3. The transmitted power cannot exceed the total base station power.

$$\sum_{k=1}^{K} \sum_{n=1}^{N} p_{k,n,t} \leq P \quad \forall t \tag{3.4}$$

From which we infer that:

$$X_{k,t} = \left\lfloor \frac{T_{k,t}}{T_f} \right\rfloor \tag{3.5}$$

The user demand is the throughput for non-real-time service, and it is the throughput and queueing delay for real-time-users. The user demand should be proportional to the throughput and inversely proportional to the delay. We define it to be:

$$D_{k,t} = N_{k,t}\left(A + \frac{B}{X_{k,t}+1}\right) \tag{3.6}$$

As $X_{k,t}$ decreases, the demand of the user is higher because it needs the allocation before a degradation of the quality of service occurs. A and B are parameters that can be tuned in order to modify the priority of real-time applications. For non-real-time users $X_{k,t}$ is ∞. $+1$ is added to $X_{k,t}$ because we don't want $D_{k,t} \to \infty$ as $X_{k,t} \to 0$; if that happens, user k will be given the priority forever which will negatively affect the scheduler.

$U_{k,t}$ should have a similar form as $D_{k,t}$. To achieve fairness, we propose to add a parameter to increase the utility of those that did not receive enough carriers to meet their requested QoS in the last frame. If the BS knows the requested data rate of the user, and due to poor channel conditions or congestion, the MS would not be assigned enough carriers to achieve its QoS; then the BS will store the difference and include it in the residue term -or dissatisfaction with the service- ($R_{k,t-1}$) during the next scheduling frame.

$$U_{k,t} = \left[M_{k,t}\left(A + \frac{B}{X_{k,t}+1}\right)\right] - R_{k,t-1} \tag{3.7}$$

If the user is in a bad channel conditions, he might not receive enough carriers to meet the requested service QoS requirements. The minus sign is used because, by decreasing the utility, we are forcing the scheduler to give more priority for users that did not receive enough scheduling in the previous frame.

M_k is a parameter that is proportional to the data rate of each user

$$M_{k,t} = \sum_{n=1}^{n=N} C_{k,n,t} M_{k,n,t} \tag{3.8}$$

where

$$\begin{cases} C_{k,n,t} = 1 & if\ user\ k\ is\ allocated\ carrier\ n \\ C_{k,n,t} = 0 & Otherwise \end{cases} \tag{3.9}$$

and

$$\begin{cases} M_{k,n,t} = 1 & 4 - QAM & R = 1/2 \\ M_{k,n,t} = 1.5 & 4 - QAM & R = 3/4 \\ M_{k,n,t} = 2 & 16 - QAM & R = 1/2 \\ M_{k,n,t} = 3 & 16 - QAM & R = 3/4 \\ M_{k,n,t} = 4 & 64 - QAM & R = 2/3 \\ M_{k,n,t} = 4.5 & 64 - QAM & R = 3/4 \end{cases} \tag{3.10}$$

Note that the value of $M_{k,n,t}$ depends on the channel condition of user k using carrier n at a scheduling time t. To know the modulation order of the receiver, the received signal power or the received signal to noise ratio is found using:

$$SNR_{k,n,t} = p_{k,n,t} \times \frac{|H_{k,n,t}|^2}{N_{k,n,t}} \tag{3.11}$$

A suitable formulation of the residue is one that stores the difference between the demand and the utility only if the demand is greater than the utility. In other words we want to compensate for the users who did not receive enough scheduling, rather than punish those that did.

$$R_{k,t} = \begin{cases} D_{k,t} - U_{k,t} & D_{k,t} - U_{k,t} > 0 \\ 0 & Otherwise \end{cases} \tag{3.12}$$

3.2 Sampling

The traditional GA uses random initial population which may limit its performance. In this section multiple techniques to select the initial population based on sampling are proposed. The search space Θ contains billions of possibilities. A system with 10 users and 1024 carriers has 10^{1024} possible carrier allocations, which make it very hard to uniformly sample from such space. The techniques used in the sampling process are divided into two categories: direct sampling and two-level sampling.

3.2.1 Direct Sampling

The direct sampling techniques are those that are directly able to sample from the search space no matter how large it is; these are the modified random start (MRS) and the Hammersley approximation.

MRS

The main idea of MRS is to generate the initial population using randomization. It is considered as a sampling technique because the generated solution is a sample from the search space. The MRS algorithm is as follows:

Algorithm 3.1 Modified Random Start

Get the number of carriers ($N_{carriers}$).
Get the number of users (N_{users}).
Get the requested throughput for each user.
for $n = 1 \to N_{carriers}$ **do**
  Calculate the received throughput for every user.
  if If all the users receive throughput more than they requested **then**
    Allocate all the remaining carriers to the users at random.
  end if
  Allocate carrier n to a random user given that he didn't receive enough throughput.
end for

Hammersley Approximation

Hammersley approximation is another method used to directly generate the initial population. For a two dimensional data, Hammersley points are represented as follows:$\left(\frac{k}{n}, \Phi_{p_1}(k)\right)$. We propose to make the sequence of $\Phi_{p_x}(k)$ indicate the carrier allocation of chromosome x. The set of Hammersley points obtained in the previous section are uniform but they are bounded between $[0, 1]$. By multiplying $\Phi_p(k)$ by K (recall that K is the number of users), one can obtain uniformly sampled points between $[0, k]$. Since the resultant value is not an integer, a floor function is needed to make the value represent the user indexes. Therefore, the "Hammersley approximation" results to:

$$\begin{cases} user\ index & = \lfloor \Phi_p(k) \times K \rfloor \\ carrier\ index & = k \end{cases} \tag{3.13}$$

Note that the following result is for a given prime base p, and it represents a single chromosome. To generate the initial population the process is repeated

$N_{chromosomes}$ times for different base values. Where $N_{chromosomes}$ is the number of chromosomes in the initial population.

3.2.2 Two Level Sampling

In this section two techniques that are based on the spirit of ordinal optimization are presented, those are the uniform sampling inspired method (USIM) and the ordinal optimization inspired method (OOIM). Since the size of the proposed system is far beyond the range of ordinal optimization, uniform sampling from a large space will take forever. In the two level sampling technique, sampling is done in two levels: first, uniform sampling is done for a much simpler search space, then the obtained samples are expanded for the larger search space.

USIM

For the reason stated above, USIM can't sample directly form the search space, instead it will use the result from uniform sampling for small system space, and then the result is generalized to get the initial carrier allocation.

Uniform Sampling For Small System Space In this model the number of users and the number of carriers are selected to be 4 and 16 respectively. The reduced search space has a total number of possibilities of $4^{16} = 4294967296$, which is within the range of OO. To uniformly sample from such population algorithm 3.2 is proposed.

Algorithm 3.2 Uniform Sampling

Select the number of samples $N_{Samples}$ {The number should be around 1000}
Set the value of K {In this case $K = 4$}
Set the value of N {In this case $N = 16$}
Generate an empty array V of length N_S
for $j = 1 \to N_S$ **do**
  Get the index $i = \frac{K^N}{j} + \epsilon$ {Where ϵ is an added random number much less than $\frac{K^N}{j}$}
  transform index i to base K
  $V[j] \leftarrow i$
  $j \leftarrow j + 1$
end for

The obtained result is an array of length $N_{Samples}$, each element contains 16 digits: $d_1 d_2 \cdots d_{16}$. Those digits signify that carrier 1 is allocated to user d_1 and carrier 2 is allocated to user d_2 and so on and so forth.

Sampling For The Original System Space To sample from a search space of size 10^{1024} ($K = 10$ and $N = 1024$), algorithm 3.3 is proposed.

The obtained result represents a feasible solution. It can be improved if GA is added to it. Note that all the steps above are only for one chromosome, the process is repeated $N_{chromosomes}$ times to generate the initial population for GA. Note that the smaller model (16 carriers and 4 users) is because we are assuming that the obtained result will have a similar distribution as the original search space. USIM can easily extrapolate for more users and for more carriers since the selection is random. However the number of carriers must be a multiple of 16, which is the size of each sub-chromosome.

Algorithm 3.3 Uniform Sampling Inspired Method

Get the requested throughput for each user.

Create dynamic vector u of length K, containing all the numbers from 1 to K in random order.

Generate the array v from algorithm 3.2 $\{v$ contains 1000 uniform samples$\}$.

Create an empty array w of length 4.

$N_{S-Ch} \leftarrow \frac{N}{16}$

for $o = 1 \rightarrow N_{S-Ch}$ **do**

  for $n = 1 \rightarrow N_{Ch}$ **do**

    Randomly shuffle v and store its first four values in w

    for $m = 1 \rightarrow length(v)$ **do**

      $A[m] = v[m]$ $\{A[m]$ contains 16 digits. The value every one of them is either 0, 1, 2 or 3$\}$.

      Replace the digits of $A[m]$ valued 0, 1, 2 and 3 by w[0], w[1], w[2] or w[3] respectively. $\{A[m]$ now contains 16 digits d_1, d_16, $\ldots d_15$. The value every one of them is either w[0], w[1], w[2] or w[3]. This signify that carrier 1 is allocated to user d_1 and carrier 2 is allocated to user d_2 and so on and so forth$\}$.

    end for

    Select the value of $A[m]$ that result to the highest throughput. Then allocate the carriers to users indicated by A by storing the result in chromosome n.

    Calculate the received throughput for every user.

    if If all the users receive throughput more than they requested **then**

      Reset the value of u $\{$To contain all intergers from 1 to $K\}$

    else

      Remove the entries from u that has the values of user ID who receive throughput more than they requested

    end if

    while length$(u) \leq 4$ **do**

      Select at random a value from u. Add another entry for u equal to the selected value.

    end while

  end for

end for

OOIM

Like USIM, OOIM also uses uniform sampling for small system space to obtain the initial carrier allocation. From algorithm 3.3, the chromosome of length 1024 is divided into 64 sub-chromosomes. Only the best sub-chromosomes are selected. The purpose of OOIM is to generate S chromosomes from only one chromosome. That is from the 64 sub-chromosomes, the top S combination is selected. Each sub-chromosome has 4^{16} possibilities, if the top 2 designs from each sub-chromosome are selected; the obtained result will have 2^{64} possibilities. Choosing the top S design form them is very time consuming. To select the top S designs the approximate algorithm 3.4 is presented:

Note that the obtained method is less computationally expensive than USIM because all the obtained chromosomes are derived from only one chromosome.

3.3 Ordinaly Optimized Evolutionary Scheduler

The steps of the evolutionary scheduler are the same as regular genetic algorithm. It is called Ordinaly Optimized because the initial population and the stopping criteria can be determined by the use of OO. The structure of "Ordinaly Optimized" evolutionary scheduler (OOES) and of OO-GA algorithm are shown in figures 3.2, and 3.3 respectively. *Note that OOES and OO-GA have the same workflow, with only difference in selecting the initial population. OOES uses OOIM as initial population sampling algorithm, while OO-GA uses all other initial population sampling algorithms.*

Algorithm 3.4 Ordinal Optimization Inspired Method

Get the requested throughput for each user.
Create dynamic vector u, same vector presented in algorithm 4.3
Generate the array v from algorithm 4.2
Create an empty array w of length 4.
$N_{S-Ch} \leftarrow \frac{N}{16}$
Create an array x of length N_{S-Ch} which has three entries valued as 1 while the others are zeros.
for $l = 1 \rightarrow N_{Ch}$ **do**
  Randomly shuffle x.
  $y[l][:] \leftarrow x[:]$
end for
for $n = 1 \rightarrow N_{S-Ch}$ **do**
  Randomly shuffle v and store its first four values in w
  for $m = 1 \rightarrow length(v)$ **do**
    $A[m] = v[m]$
    Replace the digits of $A[m]$ valued 0, 1, 2 and 3 by w[0], w[1], w[2] or w[3] respectively.
  end for
  for $l = 1 \rightarrow N_{Ch}$ **do**
    if $y[l][n] = 0$ **then**
      Select the value of $A[m]$ that results to the highest throughput.
    else
      Select the value of $A[m]$ that results to the second highest throughput.
    end if
    Allocate the carriers to users indicated by A, and store the result into chromosome l.
  end for
  Calculate the received throughput for every user.
  if If all the users receive throughput more than they requested **then**
    Reset the value of u {To contain all intergers from 1 to K}
  else
    Remove the entries from u that has the values of user ID who receive throughput more than they requested
  end if
  while length(u) ≤ 4 **do**
    Select at random a value from u. Add another entry for u equal to the selected value.
  end while
end for

Carrier ID	1	2	3	4	...	1024
User ID	1	5	6	8	...	5

Figure 3.1: Chromosome structure

The chromosome structure is shown in figure 3.1. The chromosome length is equal to the number of carriers. The index value is the carrier ID while the value pointed by the index is the user ID. As long as the user ID is $\epsilon\,[1, K]$, the chromosome will always generate a solution inside the search space.

3.3.1 Initial Selection

The initial population is chosen by using one of the techniques presented in section 3.2.

3.3.2 Mutation

Two methods of mutation are proposed:

1. The mutation occurs for each element in the chromosome with a mutation probability P_m.

2. The mutation is a swap between two user indexes for two carriers. The swap occur only if it benefits both sides.

Note that the first method may lead to unbalanced carrier allocation which can be beneficial if the user demand is unbalanced, while the second method will increase the fitness for sure, however every user will have the same number of carriers provided from the initial allocation if there is no crossover.

3.3.3 Crossover

A single point crossover is proposed, where the crossover point is selected at random for every generation and every chromosome.

3.3.4 Selection

The selection criteria chosen is the roulette wheel selection based on the highest fitness. Only the selected chromosomes are chosen for the next generation. Note that the fitness function is the same as the objective presented in section 3.1.

3.3.5 Stopping Criteria

For the stopping criteria, the algorithm presented in [31] is used. If the size of the selected set S is known, it can determine the size of the initial population ($S_{initial}$) or the stopping criteria $N_{generations}$ by the following equation:

$$S = S_{initial} \times N_{generations} \tag{3.14}$$

The value of S is equal to the value returned by OO.

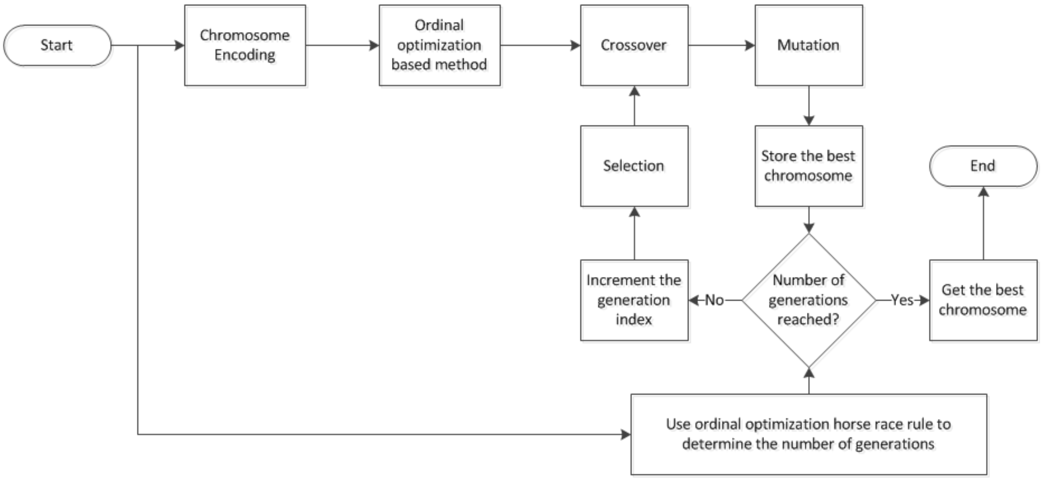

Figure 3.2: OOES structure

3.3.6 Offspring

The process repeats until convergence occurs or the number of iterations reached its limits. The main idea behind this process is that OO alone has a 95% of confidence. By adding GA to it, the selected population is expected to improve.

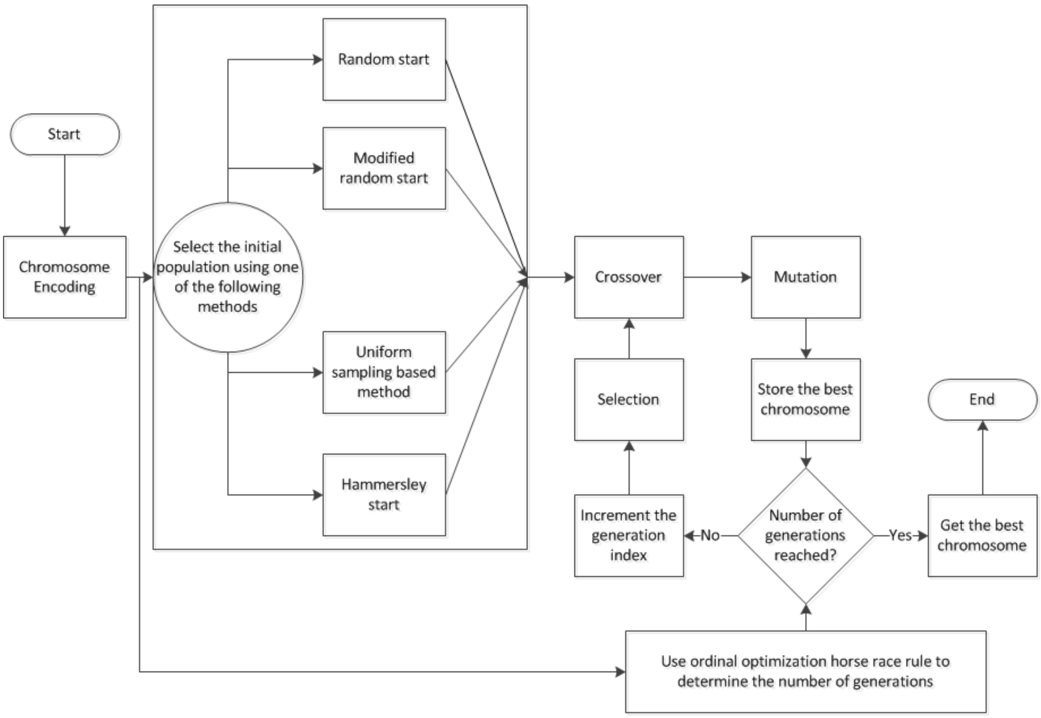

Figure 3.3: OO-GA algorithm

Chapter 4

Simulation

In this chapter, simulation is done to validate the usefulness of the work provided in the previous chapters. First, the system parameters that are used in the simulation are presented in section 4.1. Second, the proposed fitness function is compared against those that were suggested in [23, 27, 22] by using different GA operators such as initial population, mutation and crossover. Finally, complexity analysis is used to determine which algorithms are more complex than others.

4.1 System Parameters

In this section, all systems parameters that are used in the simulation are presented. First, in section 4.1.1, the system assumptions show the users demand and their requested QoS requirement. Second, different channel models are derived in section 4.1.2. Third, in section 4.1.3, OO is used to determine the number of iterations. Finally, the parameter setting are summarized in section 4.1.4.

4.1.1 System Assumptions

We consider a single cell downlink OFDMA system with 10 users and 1024 subcarriers. The users are divided into two classes: real-time and non-real-time users. The BS must perform scheduling and carrier allocation to the users by satisfying the following design specifications:

- Maximize the system throughout.
- Meet the QoS for all types of services.
- Maintain fairness among users.
- Minimize power consumption.
- Be as simple as possible (We must be able to perform scheduling within the frame period).

In the proposed system model we will assume the following parameters:

Number of users	10
OFDMA symbol duration	125 μs
OFDMA frame duration	5 ms
Number of data subcarriers	1024

Table 4.1: OFDMA parameters

The data rate is obtained by using the following formula:

$$Rate = Num_{bits/OFDMA\,symbol} \times Num_{symbols/frame} \times Num_{frames/second} \quad (4.1)$$

User id	1	2	3	4	5
Distance from base station in m	100	200	300	400	500
Requested throughput in Mbps	8	1.2	0.96	0.96	0.8
Service class (1:real-time 0:non-real-time)	0	0	0	0	0
User id	6	7	8	9	10
Distance from base station in m	100	200	300	400	500
Requested throughput in Mbps	6.4	3.2	2.32	0.88	0.88
Service class (1:real-time 0:non-real-time)	1	1	1	1	1

Table 4.2: Assumed user required rate

$$Number\ of\ bits\ per\ OFDMA\ symbol = \sum Number\ of\ bits\ per\ subcarrier \tag{4.2}$$

The data rate requested from each user is shown in table 4.2

According to [32] these data rates are more than enough rates for FTP, web browsing, enhanced web browsing and email for non-real-time users, as well as for audio conferencing, video conferencing and voice over IP.

4.1.2 Channel Model

For the OFDMA channel model we will use the modified IEEE 802.16d proposed in [33] where:

$$PL_{M-IEEE\,802.16d} = \begin{cases} 20\log_{10}\left(\frac{4\pi d}{\lambda}\right) & d \leq d_0' \\ 20\log_{10}\left(\frac{4\pi d_0'}{\lambda}\right) + 10\gamma\log_{10}\left(\frac{d}{d_0}\right) + C_f + C_{RX} & d > d_0' \end{cases} \tag{4.3}$$

With d_0 is the reference distance which is $100m$.

d_0' is the new reference distance which is given by:

$$d_0' = d_0 \times 10^{-\frac{C_F + C_{RX}}{10\gamma}} \tag{4.4}$$

C_F is the correlation coefficient for the carrier frequency $f_c[MHz]$ and it is given by:

$$C_F = 6\log_{10}\left(\frac{f_c}{2000}\right) \tag{4.5}$$

C_{RX} is the correlation coefficient for the receive antenna:

$$C_{RX} = \begin{cases} -10.8\log_{10}\left({}^{h_{RX}}/_{2}\right) & for\ Type\ A\ and\ B \\ -20\log_{10}\left({}^{h_{RX}}/_{2}\right) & for\ Type\ C \end{cases} \tag{4.6}$$

Where A, B and C are the terrain type which are shown in table 4.3.

γ is pathloss coefficient:

$$\gamma = a - bh_{TX} + {}^{c}/_{h_{TX}} \tag{4.7}$$

The values of a, b and c can be taken from table 4.4, given the terrain type.

The pathloss model is shown in the figure 4.1

As we can see, the graph is continuous. For more details and explanations

Type	Description
A	Macro-cell suburban, ART to BRT for hilly terrain with moderate-to-heavy tree densities
B	Macro-cell suburban, ART to BRT for intermediate path loss condition
C	Macro-cell suburban, ART to BRT for flat terrain with light tree densities

Table 4.3: Type of terrains [33].

Parameter	Type A	Type B	Type C
a	4.6	4	3.6
b	0.0075	0.0065	0.005
c	12.6	17.1	20

Table 4.4: a, b and c parameters for the pathloss coefficient [33].

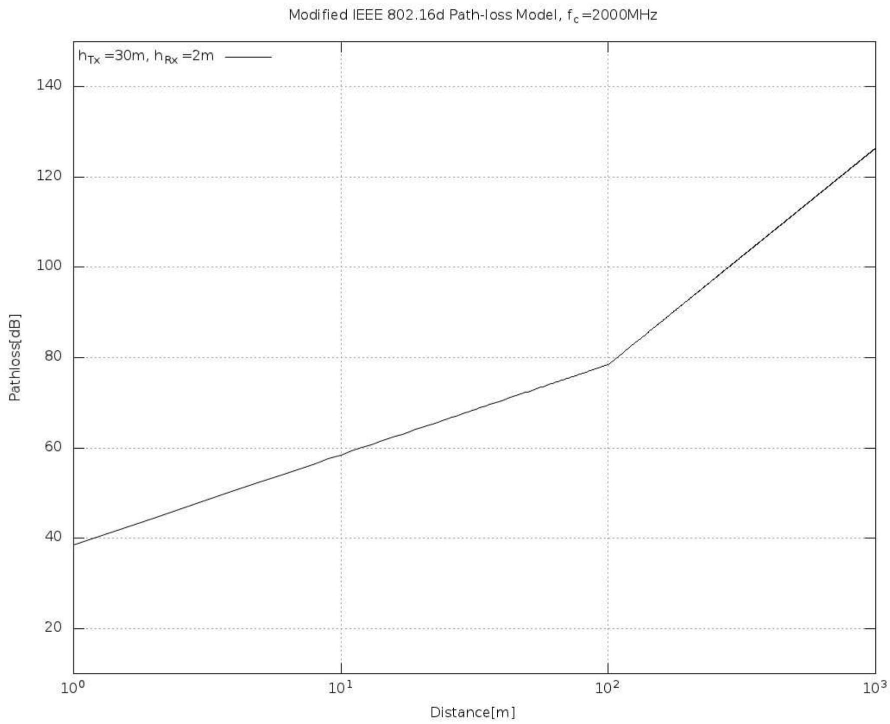

Figure 4.1: Pathloss in [dB] as a function of distance

on this path loss model please refer to [33].
We add the shadowing effect to the subcarriers in order to get a more realistic channel effect. We assume that the CSI is known at the receiver in order to perform scheduling.

4.1.3 OO

In this section OO is used to determine the number of iterations in GA.

Uniform Sampling

To uniformly sample from the smaller system, the algorithm presented in 3.2.2 is used.

Crude Model

The crude model chosen in this case is the throughput. The carrier allocation from before is fed to the model.

$$crude\,model = \sum_{n=1}^{n=N} \sum_{k=1}^{k=K} C_{k,n,t} M_{k,n,t} \tag{4.8}$$

Design Parameters

The size of the good enough set chosen is $G = 10$ and the alignment level $K = 2$

Order Performance Curve

The obtained curves are shown in figures 4.2 and 4.3:

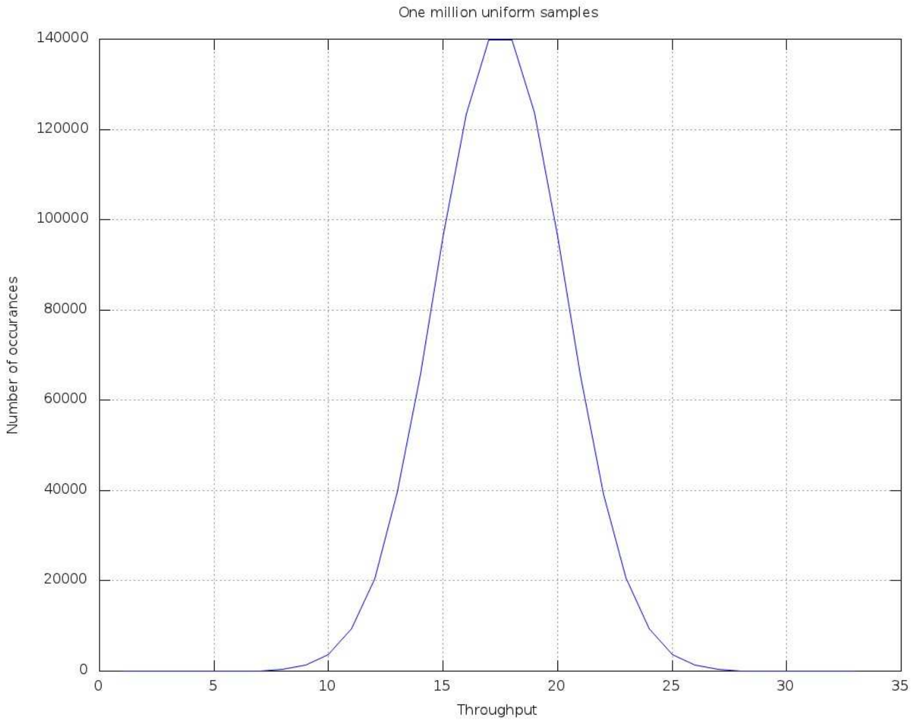

Figure 4.2: Throughput vs. number of occurrences

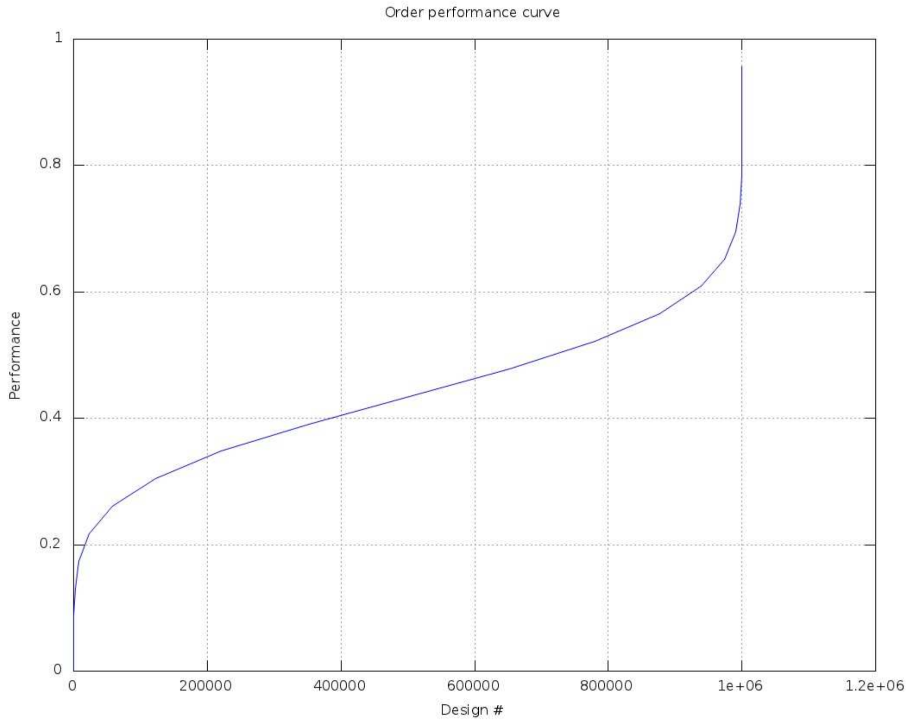

Figure 4.3: Order performance curve

Note that to get a clear look of the OPC, one million samples are chosen; in practice the number of samples chosen is around 1000.

Selected Set Size

From the OPC obtained, the most suitable curve from figure 2.2 is bell shaped (many mediocre designs). For a Bell OPC with medium noise level according to figure 2.3:

$Z_1 = 8.5988$

$Z_2 = 1.4089$

$Z_3 = -1.6789$

$Z_4 = 9.00$

By applying equation 2.2, one can obtain:

$S = e^{Z_1} K^{Z_2} G^{Z_3} + Z_4 = e^{8.5988} * 2^{1.4089} * 10^{-1.6789} + 9 = 310$

If the size of the initial population is selected to be 10, according to equation 2.2 the number of generations is 31.

4.1.4 Parameter Settings

The rest of the simulation are done using the data from table 4.2, which is the case of heavy unbalanced demand between users. *Unless mentioned otherwise* the GA parameters are those presented in table 4.5. The simulation was done under bad channel conditions. *Note that waterfilling algorithm and fitness 1 are used interchangeably for the rest of simulation. Also note that all the results are averaged over 1000 runs.*

Mutation operator	Swap if better
Random Mutation probability	0.5
"Swap if better" mutation probability	$>>$1 (Some carriers may mutate more than once)
Crossover operator	Single point, position changes randomly
Crossover probability	0.3
Selection operator	Roulette wheel
Number of generations	31
Initial population	10
Fitness function	Proposed fitness function (section 3.1, equation 3.1)

Table 4.5: GA parameters

4.2 Effect of GA Operators

In this section, the effect of GA operators such as fitness function, mutation, crossover and initial population on the carrier allocation will be tested. The average results for 1000 runs are reported in tables 4.6, 4.7 and 4.8 and figures 4.4, 4.5, 4.6 and 4.7 under bad channel conditions. Note that for the rest of the simulation, fitnesses 1, 2 and 3 refer to equations 2.6, 2.8 and 2.11 respectively.

4.2.1 Effect of Mutation Operator

In this section the effect of mutation operator is tested for GA, with different initial population, and different fitness functions under bad channel conditions.

As can be seen from table 4.6, the carrier allocation with a "swap if better" mutation always leads to higher throughput than the carrier allocation with random mutation. On the other hand, the carrier allocation with

Fitness, algorithm, Mutation	Total Throughput	Standard deviation
Proposed, OOES (OOIM), Swap	28734398	235880
Proposed, OO-GA (USIM), Swap	28948992	343683
Proposed, OO-GA (MRS), Swap	28041552	176759
Proposed, OO-GA (Hammersley), Swap	27638444	156096
Fitness 1, OOES (OOIM), Swap	28537976	260879
Fitness 1, OO-GA (USIM), Swap	28578508	355359
Fitness 1, OO-GA (MRS), Swap	27800932	224705
Fitness 2, OOES (OOIM), Swap	27440704	205793
Fitness 2, OO-GA (USIM), Swap	27443284	209967
Fitness 2, OO-GA (MRS), Swap	27448924	204772
Fitness 3, OOES (OOIM), Swap	28507716	258984
Fitness 3, OO-GA (USIM), Swap	27432144	212115
Fitness 3, OO-GA (MRS), Swap	27716384	225948
Proposed, OOES (OOIM), Random	26306792	175811
Proposed, OO-GA (USIM), Random	26319244	165225
Proposed, OO-GA (MRS), Random	25280816	73250
Fitness 1, OOES (OOIM), Random	23179900	104948
Fitness 1, OO-GA (USIM), Random	23183204	102639
Fitness 1, OO-GA (MRS), Random	23184792	102317
Fitness 2, OOES (OOIM), Random	22724812	177086
Fitness 2, OO-GA (USIM), Random	22722504	175824
Fitness 2, OO-GA (MRS), Random	22721848	173804
Fitness 3, OOES (OOIM), Random	22912244	177983
Fitness 3, OO-GA (USIM), Random	22899004	173734
Fitness 3, OO-GA (MRS), Random	22897264	175200

Table 4.6: Effect of fitness function, initial population and mutation operator on the total throughput in bps under bad channel conditions, for mutation probability $>>$1.

Fitness, Selection, Mutation	% of under-allocated users	Standard deviation
Proposed, OOES (OOIM), Swap	10.25	6.82
Proposed, OO-GA (USIM), Swap	10.97	8.40
Proposed, OO-GA (MRS), Swap	10.12	8.05
Proposed, OO-GA (Hammersley), Swap	18.48	7.93
Fitness 1, OOES (OOIM), Swap	17.25	8.09
Fitness 1, OO-GA (USIM), Swap	19.67	8.64
Fitness 1, OO-GA (MRS), Swap	20.26	7.84
Fitness 2, OOES (OOIM), Swap	25.74	7.55
Fitness 2, OO-GA (USIM), Swap	25.18	7.68
Fitness 2, OO-GA (MRS), Swap	25.57	7.62
Fitness 3, OOES (OOIM), Swap	17.59	8.47
Fitness 3, OO-GA (USIM), Swap	25.55	7.16
Fitness 3, OO-GA (MRS), Swap	21.71	7.93
Proposed, OOES (OOIM), Random	20.67	10.28
Proposed, OO-GA (USIM), Random	14.81	4.99
Proposed, OO-GA (MRS), Random	29.43	2.31
Fitness 1, OOES (OOIM), Random	41.68	4.62
Fitness 1, OO-GA (USIM), Random	42.05	4.55
Fitness 1, OO-GA (MRS), Random	41.91	4.78
Fitness 2, OOES (OOIM), Random	33.37	5.43
Fitness 2, OO-GA (USIM), Random	33.49	5.43
Fitness 2, OO-GA (MRS), Random	33.25	5.28
Fitness 3, OOES (OOIM), Random	38.92	3.85
Fitness 3, OO-GA (USIM), Random	38.67	4.09
Fitness 3, OO-GA (MRS), Random	38.71	3.85

Table 4.7: Effect of fitness function, initial population and mutation operator on the percentage of under-allocated users under bad channel conditions, for mutation probability $>>1$.

Fitness, Selection, Mutation	% of users under-allocated by >10%	Standard deviation
Proposed, OOES (OOIM), Swap	2.76	5.42
Proposed, OO-GA (USIM), Swap	1.83	4.11
Proposed, OO-GA (MRS), Swap	0.44	2.05
Proposed, OO-GA (Hammersley), Swap	2.33	4.25
Fitness 1, OOES (OOIM), Swap	4.63	5.73
Fitness 1, OO-GA (USIM), Swap	6.69	6.99
Fitness 1, OO-GA (MRS), Swap	4.16	5.75
Fitness 2, OOES (OOIM), Swap	6.08	6.51
Fitness 2, OO-GA (USIM), Swap	5.82	6.44
Fitness 2, OO-GA (MRS), Swap	6.06	6.68
Fitness 3, OOES (OOIM), Swap	4.54	5.56
Fitness 3, OO-GA (USIM), Swap	5.77	6.29
Fitness 3, OO-GA (MRS), Swap	4.21	5.67
Proposed, OOES (OOIM), Random	11.87	4.42
Proposed, OO-GA (USIM), Random	9.42	3.50
Proposed, OO-GA (MRS), Random	7.05	4.56
Fitness 1, OOES (OOIM), Random	28.33	5.99
Fitness 1, OO-GA (USIM), Random	28.6	6.20
Fitness 1, OO-GA (MRS), Random	28.69	5.76
Fitness 2, OOES (OOIM), Random	27.96	4.08
Fitness 2, OO-GA (USIM), Random	28.39	3.81
Fitness 2, OO-GA (MRS), Random	28.26	3.97
Fitness 3, OOES (OOIM), Random	28.13	5.08
Fitness 3, OO-GA (USIM), Random	27.93	5.00
Fitness 3, OO-GA (MRS), Random	28.16	5.20

Table 4.8: Effect of fitness function, initial population and mutation operator on the percentage of highly under-allocated users under bad channel conditions, for mutation probability >>1.

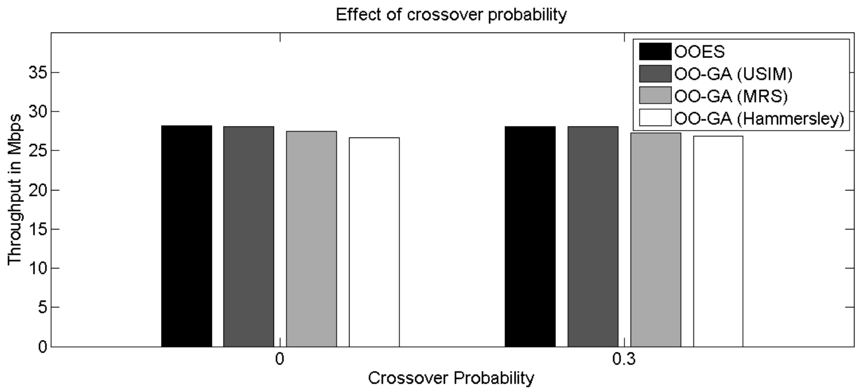

Figure 4.4: Effect of initial population and crossover probability on the total throughput, for the proposed fitness using "swap if better" mutation, under bad channel conditions, for mutation probability = 0.5.

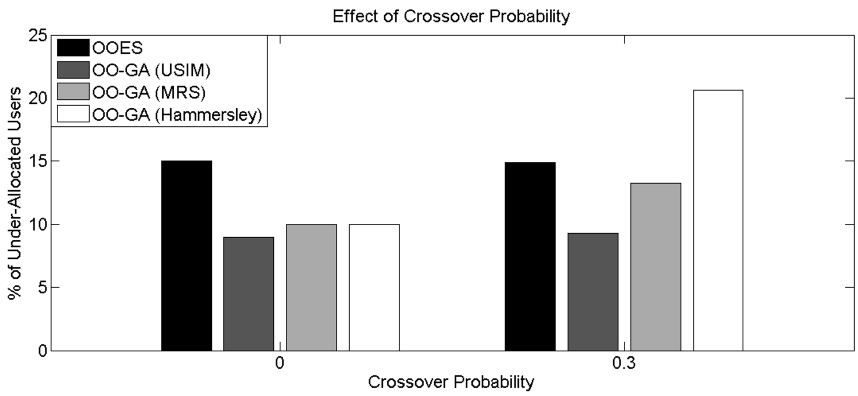

Figure 4.5: Effect of initial population and crossover probability on the percentage of under-allocated users, for the proposed fitness using "swap if better" mutation, under bad channel conditions, for mutation probability = 0.5.

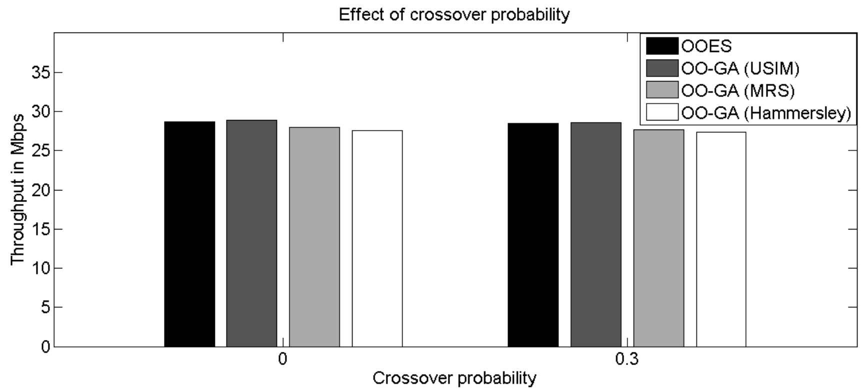

Figure 4.6: Effect of initial population and crossover probability on the total throughput, for the proposed fitness using "swap if better" mutation, under bad channel conditions, for mutation probability >>1.

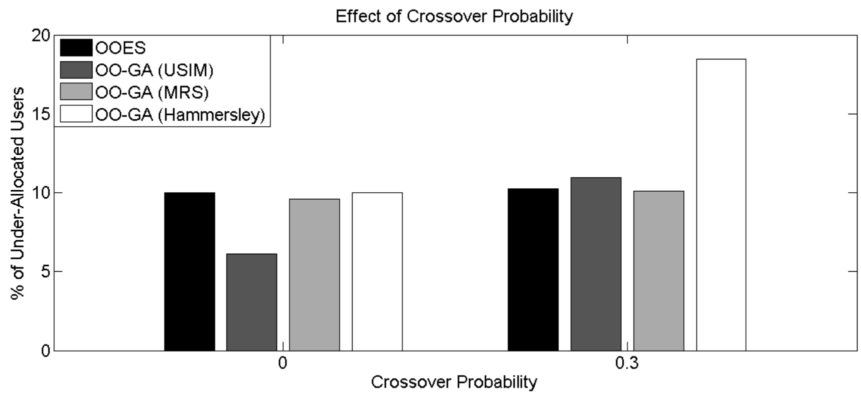

Figure 4.7: Effect of initial population and crossover probability on the percentage of under-allocated users, for the proposed fitness using "swap if better" mutation, under bad channel conditions, for mutation probability >> 1.

random mutation has lower standard deviation. In terms of the percentage of under-allocated users and highly under-allocated users, as shown in tables 4.7 and 4.8, the "swap if better" mutation has lower percentage of users who didn't receive their requested QoS. Overall, the "swap if better" mutation leads to a better carrier allocation than the random mutation if they are tested under the same conditions.

4.2.2 Effect of Fitness Function

In this section the effect of fitness function is tested for GA with different initial population, and different mutation operators under bad channel condition.

In the case of "swap if better" mutation, in terms of total throughput, as can be seen from table 4.6, the proposed fitness function reached the highest throughput under all initial population algorithms while fitness 1 allocate with the second highest throughput. In terms of the percentage of under-allocated users the proposed fitness carrier allocation leads to the least percentage of dissatisfied users under all initial population algorithms, which is about 10%. The other algorithms under-allocate at least 17% of the users for different initial population algorithms. In terms of the percentage of the users who were under-allocated by more than 10%, the proposed fitness carrier allocation leads to the least percentage of high dissatisfied users under all initial population algorithms, which is less than 2.8%. The other algorithms carrier allocation leads to at least 4.2% of highly dissatisfied users for different initial population algorithms. Overall, the proposed fitness function carrier allocation leads to higher throughput and less percentage of under-allocated users than other fitness functions if it is tested under the same conditions. Note that the same conclusions can be drawn if the comparison is done for random mutation.

4.2.3 Effect of Initial Population

In this section the effect of initial population is tested for GA using the proposed fitness with different initial population, with "swap if better" mutation under bad channel condition. In terms of total throughput the USIM algorithm attained the highest throughput of 28948992 bps while the Hammersley algorithm reached the lowest throughput of 27638444 bps. In terms of standard deviation the MRS allocation leads to the lowest throughput standard deviation of 156096 while the USIM leads to the highest standard deviation of 343683 bps. In terms of user under-allocation, except for Hammersley approximation, all algorithms performed similarly by under-allocating about 10% of the users. In terms of users who were under-allocated by more than 10% all algorithms under-allocate less than 3% of the users. In addition, the percentage of users who were under-allocated by more than 10% using MRS are near zero. To conclude, the MRS performed better than other algorithms because it is much simpler than USIM and OOIM.

Another way to test the effect of initial population on the proposed fitness function is to lower the probability of "swap if better" mutation to 50%. The reason behind it is to avoid getting "good solution" from hill climbing. In this way the effect of initial population is better tested. The results are shown in figures 4.4 and 4.5. From figure 4.5, it is noticeable that USIM is much better than MRS and OOIM in terms of the percentage of under-allocated users since it acheives under-allocation percentage less than 9%. Regarding total throughput, OOES attained the highest throughput of 28297808 bps while Hammersley attained the lowest of throughput of 26778996 bps.

4.2.4 Effect of Crossover

In this section the effect of crossover is tested for GA using the proposed fitness, different initial population, different mutation probability, and using "swap if better" mutation under bad channel condition. The results are shown in figures 4.4, 4.5, 4.6 and 4.7. By examining figures 4.4 and 4.6, it can be noticed that the crossover probability has very little effect on the the total throughput whether is the mutation probability is high (figure 4.6) or low (figure 4.4). On the other hand, the effect of crossover is very clear by examining figures 4.5 and 4.7. The percentage of under-allocated users increases for GA using different initial population by the increase of crossover probability. This is because of the following: If the initial population selection algorithm allocates "good" number of carriers for each user, the "swap if better" mutation operator will generate better results while the number of carriers for each user remains the same; then, the crossover operator will modify the number of carriers allocated for each user. This change will negatively affect the percentage of under-allocated users.

4.3 Complexity Analysis

In addition to the performance of the scheduling algorithms, complexity analysis is also very important. It determines if the increase in complexity would lead to much better results or the increase in performance would be so small that a simpler algorithm would behave similarly with much less time. To perform the complexity analysis, a "C profiler" tool is needed. The simulation is done by using Intel centrino 2 processor with 3GB RAM. The most important outcomes of the C profiler are:

1. ms/Call: it represents the total time required to execute the function (including all the nested functions)

2. # of calls: it indicates the number of time the functions was called during the execution of the program

3. % Time: it indicates the percentage of time it takes to run the specified function.

By examining table 4.9, it can be noticed that except for USIM, the "swap if better" mutation is taking the highest amount of time to compute. That is because of the large probability of mutation. If the mutation probability is decreased, the algorithm will take less time. Since USIM is very complex it is therefore an unpractical solution from computation time perspective. Although OOIM is less complex, it still is computationally expensive: if parallel implementation is performed OOIM would perform much better. Hammersley and MRS are relatively simple to implement, therefore they are practical. The main drawback of MRS and Hammersley from is that

Function	ms/Call	# of calls	% Time
OOIM	224.93	1	39.25
"Swap if better" mutation (OOIM)	$\simeq 0$	15500000	42.24
Total time (OOES)	0.573		
USIM	128.35	10	78.02
"Swap if better" mutation (USIM)	$\simeq 0$	15500000	16.29
Total time (GA+USIM)	1.645		
MRS	0.02	10	$\simeq 0$
"Swap if better" mutation (MRS)	$\simeq 0$	15500000	74.87
Total time (GA+MRS)	0.35		
OO-GA (Hammersley)	0.1	10	$\simeq 0$
"Swap if better" mutation (Hammersley)	$\simeq 0$	15500000	79.45
Total time (GA+Hammersley)	0.394		

Table 4.9: Complexity analysis of the initial population selection algorithms and the "swap if better" mutation, under bad channel conditions, and for very high probability of mutation.

Function	Total time for OO-GA (MRS) in ms
Proposed	30
Fitness 1	25
Fitness 2	21
Fitness 3	16

Table 4.10: Complexity analysis of OO-GA (MRS) using different fitness functions with low probability of mutation

they are sequential, therefore parallel implementation will not speed up these algorithms. Note that with the high probability of mutation all algorithms fail to meet the 20 ms time constraints.

By examining table 4.10, it can be noticed that with 50% probability of mutation, the algorithms are taking much less time to compute. Fitness 3 is the only one that is able to meet the 20 ms time constraints. Fitness 2 is very close to satisfy those constraints. While the proposed fitness and fitness 1 are a little far. Note that if a higher-performance processor is used, all algorithms would be able to meet the time constraints. Otherwise, fitness 3 would be the only feasible solution.

Chapter 5

Conclusions

In this thesis, a solution for the scheduling and resource allocation problem in the downlink of OFDMA system is proposed by the use of OO-GA and OOES. The extremely large space is sampled by the use of approximation to obtain a good initial population. Then by the use of evolution, a better carrier allocation can be obtained. The proposed fitness used the throughput and delay as metrics. It also added a compensation factor for users who did not achieve enough carriers in the current scheduling frame. In that way it can maximize the fairness while not jeopardizing the total throughput. The simulation leads to the following conclusions:

- The use of ordinal optimization can generate a good solution. The carrier allocation can be further optimized if OO is combined with GA.
- The use of "swap if better" mutation can benefit GA to achieve better allocation.
- The initial population is the parameter that has the highest effect on

GA. It presents a tradeoff between performance and complexity.

- OOES performed similar to OO-GA with USBM, even-though the former is much simpler.

- The crossover operation can be beneficial in some cases and harmful in others. It depends on the algorithm used to sample the initial population.

- Fitness function has a major impact on scheduling. It offers tradeoff between throughput maximization and meeting the QoS requirements.

For future work we will consider the following:

- Parallel implementation of the proposed algorithm to see if it can manage to perform scheduling before the time duration expires.

- Optimize the fitness function to be able to meet the QoS requirements for all users under bad channel condition. This might be done by adding another compensation factor, but this time from between generations and not from frame to frame.

- Modify the OOES to make it suitable for uplink.

- Test the proposed method on standardized system such as WiMAX or LTE with MIMO, and on a real user scenario by simulate the channel model using dedicated software tool.

- Test the effect of modifying the GA operators, such as mutation probability, crossover operator and selection criteria using the following modifications: first by making the mutation probability variable in function of number of iteration instead of a predefined static value, second by using different crossover operators, and finally by utilizing selection criteria other than roulette wheel.

Appendix A

Abbreviations

3GPP	Third Generation Partnership Project
BP	Blind Pick
BS	Base Station
CSI	Channel State Information
GA	Genetic Algorithm
HR	Horse Race
ISI	Intersymbol Interference
LTE	Long-Term Evolution
MIMO	Multiple Input Multiple Output
MRS	Modified Random Start
MS	Mobile Station
OFDM	Orthogonal Frequency Division Multiplexing
OFDMA	Orthogonal Frequency Division Multiple Access
OO	Ordinal Optimization
OOES	Ordinaly Optimized Evolutionary Scheduler

OOIM	Ordinal Optimization Inspired Method
OPC	Order Performance Curve
QoS	Quality of Service
SNR	Signal-to-Noise Ratio
SS	Subscriber Station
UAP	Universal Alignment Probability
USIM	Uniform Sampling Inspired Method
VOIP	Voice Over IP
WiMAX	Worldwide interoperability for Microwave Access

Bibliography

[1] "http://www.cisco.com/en/us/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.html,"

[2] L. Wang, L. Zhang, and D. Zheng, "A class of order-based genetic algorithm for flow shop scheduling," *The International Journal of Advanced Manufacturing Technology*, vol. 22, no. 11, pp. 828–835, 2003.

[3] Y. Ho, R. Sreenivas, and P. Vakili, "Ordinal optimization of deds," *Discrete Event Dynamic Systems*, vol. 2, no. 1, pp. 61–88, 1992.

[4] T. Lau and Y. Ho, "Universal alignment probabilities and subset selection for ordinal optimization," *Journal of Optimization Theory and Applications*, vol. 93, no. 3, pp. 455–489, 1997.

[5] Y. Ho, Q. Zhao, and Q. Jia, *Ordinal optimization: Soft optimization for hard problems*, vol. 16. Springer-Verlag New York Inc, 2007.

[6] J. Holland and J. Reitman, "Cognitive systems based on adaptive algorithms," *ACM SIGART Bulletin*, no. 63, pp. 49–49, 1977.

[7] T. Wong, W. Luk, and P. Heng, "Sampling with hammersley and halton points," *Journal of Graphics Tools*, vol. 2, no. 2, pp. 9–24, 1997.

[8] H. Huang, Y. Chen, X. Tong, and W. Wang, "Incremental wavelet importance sampling for direct illumination," in *Proceedings of the 2007 ACM symposium on Virtual reality software and technology*, pp. 149–152, ACM, 2007.

[9] J. G. Andrews, A. Ghosh, and R. Muhamed, *Fundamentals of WiMAX: Understanding Broadband Wireless Networking (Prentice Hall Communications Engineering and Emerging Technologies Series).* Prentice Hall PTR, Mar. 2007.

[10] C. So-In, R. Jain, and A. Tamimi, "Scheduling in ieee 802.16 e mobile wimax networks: key issues and a survey," *Selected Areas in Communications, IEEE Journal on*, vol. 27, no. 2, pp. 156–171, 2009.

[11] X. Xie, H. Chen, and H. Wu, "Simulation studies of a Fair and Effective Queueing algorithm for WiMAX resource allocation," in *Communications and Networking in China, 2008. ChinaCom 2008. Third International Conference on*, pp. 281–285, IEEE, 2008.

[12] G. Prasath, C. Fu, and M. Ma, "QoS scheduling for group mobility in WiMAX," in *Communication Systems, 2008. ICCS 2008. 11th IEEE Singapore International Conference on*, pp. 1663–1667, IEEE, 2009.

[13] A. Belghith and L. Nuaymi, "Comparison of WiMAX scheduling algorithms and proposals for the rtPS QoS class," in *Wireless Conference, 2008. EW 2008. 14th European*, pp. 1–6, IEEE, 2008.

[14] S. Khemiri, G. Pujolle, K. Boussetta, and N. Achir, "A Combined MAC and Physical Resource Allocation Mechanism in IEEE 802.16 e Net-

works," in *Vehicular Technology Conference (VTC 2010-Spring), 2010 IEEE 71st*, pp. 1–5, IEEE, 2010.

[15] X. Yang, M. Venkatachalam, and S. Mohanty, "Exploiting the MAC layer flexibility of WiMAX to systematically enhance TCP performance," in *Mobile WiMAX Symposium, 2007. IEEE*, pp. 60–65, IEEE, 2007.

[16] R. Jayaparvathy and S. Geetha, "Resource allocation and game theoretic scheduling with dynamic weight assignment in IEEE 802.16 fixed broadband wireless access systems," in *Performance Evaluation of Computer and Telecommunication Systems, 2008. SPECTS 2008. International Symposium on*, pp. 217–224, IEEE, 2009.

[17] R. Garroppo, S. Giordano, and D. Iacono, "Radio-aware scheduler for wimax systems based on time-utility function and game theory," in *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, 30 2009.

[18] D. Niyato and E. Hossain, "QoS-aware bandwidth allocation and admission control in IEEE 802.16 broadband wireless access networks: A non-cooperative game theoretic approach," *Computer Networks*, vol. 51, no. 11, pp. 3305–3321, 2007.

[19] D. Niyato and E. Hossain, "Radio resource management games in wireless networks: an approach to bandwidth allocation and admission control for polling service in IEEE 802.16 [Radio Resource Management

and Protocol Engineering for IEEE 802.16],” *IEEE Wireless Communications*, vol. 14, no. 1, pp. 27–35, 2007.

[20] E. Yaacoub and Z. Dawy, “Achieving the Nash bargaining solution in OFDMA uplink using distributed scheduling with limited feedback,” *AEU-International Journal of Electronics and Communications*, 2010.

[21] J. Song, J. Li, and C. Li, “A cross-layer wimax scheduling algorithm based on genetic algorithm,” in *Communication Networks and Services Research Conference, 2009. CNSR'09. Seventh Annual*, pp. 292–296, IEEE, 2009.

[22] Y. Teng, Y. Zhang, M. Song, Y. Dong, and L. Wang, “Genetic algorithm based adaptive resource allocation in ofdma system for heterogeneous traffic,” in *Personal, Indoor and Mobile Radio Communications, 2009 IEEE 20th International Symposium on*, pp. 2060–2064, IEEE.

[23] Y. Wang, F. Chen, and G. Wei, “Adaptive subcarrier and bit allocation for multiuser ofdm system based on genetic algorithm,” in *Communications, Circuits and Systems, 2005. Proceedings. 2005 International Conference on*, vol. 1, pp. 242–246, IEEE, 2005.

[24] H. Cheng and W. Zhuang, “Novel packet-level resource allocation with effective qos provisioning for wireless mesh networks,” *Wireless Communications, IEEE Transactions on*, vol. 8, no. 2, pp. 694–700, 2009.

[25] H. Cheng and W. Zhuang, “Joint power-frequency-time resource allocation in clustered wireless mesh networks,” *Network, IEEE*, vol. 22, no. 1, pp. 45–51, 2008.

[26] Y. Yu and W. Zhou, "Resource allocation for ofdma system based on genetic algorithm," in *Cross Layer Design, 2007. IWCLD'07. International Workshop on*, pp. 65–69, IEEE, 2007.

[27] M. Mehrjoo, S. Moazeni, and X. Shen, "A new modeling approach for utility-based resource allocation in ofdm networks," in *Communications, 2008. ICC'08. IEEE International Conference on*, pp. 337–342, IEEE, 2008.

[28] Y. Chiu, C. Chang, K. Feng, and F. Ren, "Ggra: A feasible resource-allocation scheme by optimization technique for ieee 802.16 uplink systems," *Vehicular Technology, IEEE Transactions on*, vol. 59, no. 3, pp. 1393–1401, 2010.

[29] L. Lee and Y. Fang, "Developing a self-learning adaptive genetic algorithm," in *Intelligent Control and Automation, 2000. Proceedings of the 3rd World Congress on*, vol. 1, pp. 619–624, IEEE, 2002.

[30] M. Deng and Y. Ho, "Iterative ordinal optimization and its applications," in *Decision and Control, 1997., Proceedings of the 36th IEEE Conference on*, vol. 4, pp. 3562–3567, IEEE, 1997.

[31] Y. Luo, M. Guignard, and C. Chen, "A hybrid approach for integer programming combining genetic algorithms, linear programming and ordinal optimization," *Journal of Intelligent Manufacturing*, vol. 12, no. 5, pp. 509–519, 2001.

[32] Y. Chen, T. Farley, and N. Ye, "Qos requirements of network applications on the internet," *Information Knowledge Systems Management*, vol. 4, no. 1, p. 55, 2003.

[33] Y. Cho, J. Kim, W. Yang, and C. Kang, *MIMO-OFDM Wireless Communications with MATLAB.* John Wiley & Sons, 2010.