A novel Genetic Algorithm for Adaptive Resource Allocation in Multiuser OFDM Systems with Proportional Rate Constraint

Nitin Sharma, Dr. K R Anupama
BITS Pilani Goa campus/EEE&I, Goa, India
Email: nitinn, anupkr @bits-goa.ac.in

***Abstract*—In this paper we present a novel technique for resource allocation in Multiuser Orthogonal Frequency Division Multiplexing (MU-OFDM) Systems with proportional rate constraint. The subcarrier allocation is tackled using a novel algorithm which combines the aspects of both deterministic and genetic algorithms (GA). For the calculated subcarrier allocation, bit-loading is done using the standard Water-filling Algorithm. This modified GA gave very encouraging results as can be seen from the simulation results shown. The simulation results show a marked improvement in the performance of the algorithm as the number of users increase. The capacity attained from the subcarrier allocation scheme generated by our algorithm is found to be comparable to that attained by previous algorithms.**

***Index Terms*—Genetic Algorithm, OFDM, Proportional Rate, Adaptive Resource Allocation.**

I. INTRODUCTION

The advent of new generation of communication technologies has ushered in an era of high data rates and better reliability. MU-OFDM [1] is a promising modulation technique which mitigates the effect of frequency selective fading and combat inter-symbol interference which is inherent in high data rate environments [2]. MU-OFDM is used in wireless LANs, and can support high data rate transmission. It can also be used for multiple accesses. Resource allocation in multiuser OFDM includes subcarrier allocation, power allocation, and bit loading. Developing efficient resource management techniques for such a setup has drawn enormous attention in recent years.

Solution to the resource allocation problem in multiuser OFDM has been broadly divided into two categories: Margin Adaptive (MA) and Rate Adaptive (RA). Resource allocation was tackled in [3] using margin-adaptive scheme, wherein an iterative subcarrier and power allocation algorithm was proposed to minimize the total transmit power given a set of fixed user data rates and the Bit Error Rate (BER) requirements. In [4], rate adaptive method was used wherein the objective was to maximize the total data rates over all users, subject to power and BER constraint. It was shown in [4] that in order to maximize the total capacity each subcarrier should be assigned to the user with best gain on it. The MA Optimization technique has been dealt with efficiently in [5], [6]. Genetic algorithm has been used here for resource allocation and the algorithm has shown better results than normal iterative algorithms. In [7] it was shown that RA optimization can be solved sub-optimally by separating Subcarrier allocation and Bit Loading. The RA optimization problem is a mixed binary integer programming problem. In [8] the proportional rate constraint is added to the existing RA optimization problem. However, the introduction of this constraint makes the optimization problem non-linear thus increasing the difficulty in finding the optimal solution because the feasible set is not convex.

Rate maximization and satisfying total power constraints are two seemingly conflicting objectives [9] with a lot of trade-offs. To simplify the problem, both of them are dealt separately. In this paper, we propose using a modified GA and Water-filling algorithm to optimize the RA. The proposed GA is used to generate the subcarrier allocation assuming equal power to all users. After the subcarrier allocation, the bit loading can be performed using the water filling algorithm. The inspiration for the genetic approach comes from the fact that GAs are ideal for optimization problems with a large feasible solution space where a quick sub-optimal solution will suffice. Also, the fact that GAs is seldom used to solve RAs further deepened our inquisitiveness to explore this option.

II. PROBLEM FORMULATION

Consider a multiuser OFDM system with K users and N subcarriers. The base station needs to assign a set of subcarriers to each user (subcarrier allocation) and also decide upon the number of bits per OFDM symbol to be transmitted through each subcarrier (bit allocation).Each user’s data are distributed across the set of subcarriers assigned to the user. Adaptive modulation takes place at each subcarrier based upon the number of bits assigned to it. The assumption is that each subcarrier is uniquely assigned to a single user and two or more users never share the same subcarrier.

Let R_k be the data rate of the k^{th} user. Each of the user's bits are then modulated into N, M-level QAM symbols, which are subsequently combined using the IFFT into an OFDM symbol. This is then transmitted through a slowly time-varying, frequency-selective Rayleigh channel with a bandwidth B. The subcarrier allocation is made known to all the users through a control channel; hence each user needs only to decode the bits on their assigned subcarriers.

It was assumed that each user experiences independent fading and the channel gain of user k in subcarrier n is denoted as $g_{k,n}$ with additive white Gaussian noise (AWGN) $\sigma^2 = N_o \frac{B}{N}$ where N_o is the noise power spectral density. The corresponding subchannel signal-to-noise ratio (SNR) is thus denoted as $h_{k,n} = g^2_{k,n} / \sigma^2$ and the *kth* user's received SNR on subcarrier n is $\gamma_{k,n} = p_{k,n}h_{k,n}$. The slowly time-varying assumption is crucial since it was also assumed that each user is able to estimate the channel perfectly and these estimates were made known to the transmitter via a dedicated feedback channel. These channel estimates were then used as input to the resource allocation algorithms. In order that the BER constraints be met, the effective SNR was adjusted accordingly. The BER of a square M-level QAM [10] with Gray bit mapping as a function of received SNR $\gamma_{k,n}$ and number of bits $r_{k,n}$ can be approximated to within 1 dB for $r_{k,n} \geq 4$ and BER $\leq 10^{-3}$ as

$$BER_{MQM}\left(\gamma_{k,n}\right) \approx 0.2\exp\left[\frac{-1.6\gamma_{k,n}}{2^{r_{k,n}}-1}\right] \quad (1)$$

Solving for $r_{k,n}$, we have

$$r_{k,n} = \log_2(1+\frac{\gamma_{k,n}}{\Gamma}) = \log_2(1+p_{k,n}H_{k,n}) \quad (2)$$

where $\Gamma \approx$ -ln(5BER)/1.6 is a constant SNR gap, and $H_{k,n} \approx h_{k,n} / \Gamma$ is the effective sub channel SNR.

The objective function of resource allocation problem with the proportional rate constraint is formulated as

$$\max_{c_{k,n}p_{k,n}} \frac{B}{N}\sum_{k=1}^{K}\sum_{n=1}^{N} c_{k,n}\log_2(1+p_{k,n}H_{k,n}) \quad (3)$$

Subject to:

$$C1: c_{k,n} \in \{0,1\}\ \forall k,n$$

$$C2: p_{k,n} \geq 0 \forall k,n$$

$$C3: \sum_{k=1}^{K} c_{k,n} = 1 \forall n$$

$$C4: \sum_{k=1}^{K}\sum_{n=1}^{N} c_{k,n}p_{k,n} \leq P_{tot}$$

$$C5: R_i : R_j = \Phi_i : \Phi_j \forall i, j \in \{1,....K\}, i \neq j$$

Where $c_{k,n}$ is the subcarrier allocation indicator such that $c_{k,n}$= 1 if and only if subcarrier *n* is assigned to user *k*, and P_{tot} is the transmit power constraint.

In C5,

$$R_k = \frac{B}{N}\sum_{k=1}^{K} c_{k,n}r_{k,n} \quad (4)$$

is the total data rate for user *k* and $\Phi_1:\Phi_2:..:\Phi_k$ are the normalized proportionality constants where $\sum_{k=1}^{K}\Phi_k = 1$.

Note that constraints C1 and C2 ensure the correct values for the subcarrier allocation indicator and the power respectively. C3 imposes the restriction that each subcarrier can only be assigned to one user, and C4 and C5 are the power and proportional rate constraints respectively.

III. GENETIC ALGORITHM

A genetic algorithm (GA) [11, 12, 13] is an adaptive heuristic search technique used in computing to find exact or approximate solutions to optimization and search problems. Genetic algorithms are categorized as global search heuristics. Genetic algorithms are a particular class of evolutionary algorithms (also known as evolutionary computation) that use techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover.

Genetic algorithms are implemented as a computer simulation in which a population of abstract representations (called chromosomes or the genotype of the genome) of candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem evolves toward better solutions. Traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible. The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. If the algorithm has terminated due to a maximum number of generations, a satisfactory solution may or may not have been reached.

A typical genetic algorithm requires:

- A genetic representation of the solution domain.
- A fitness function to evaluate the solution domain.

A standard representation of the solution is as an array of bits. Arrays of other types and structures can be used in essentially the same way. The main property that makes these genetic representations convenient is that their parts are easily aligned due to their fixed size, which facilitates simple crossover operation. Variable length representations may also be used, but crossover implementation is more complex in this case. Tree-like representations are explored in Genetic programming and graph-form representations are explored in Evolutionary programming.

The fitness function is defined over the genetic representation and measures the quality of the represented solution. The fitness function is always problem dependent. For instance, in the knapsack problem one wants to maximize the total value of objects that can be put in a knapsack of some fixed capacity. A representation of a solution might be an array of bits, where each bit represents a different object, and the value of the bit (0 or 1) represents whether or not the object is in the knapsack. Not every such representation is valid, as the size of objects may exceed the capacity of the knapsack. The *fitness* of the solution is the sum of values of all objects in the knapsack if the representation is valid or 0 otherwise. In some problems, it is hard or even impossible to define the fitness expression, in these cases, interactive genetic algorithms are used.

Once we have the genetic representation and the fitness function defined, GA proceeds to initialize a population of solutions randomly, and then improve it through repetitive application of mutation, crossover, inversion and selection operators. Steps involved in a Genetic algorithm (GA) are as follows:

A. Initialization

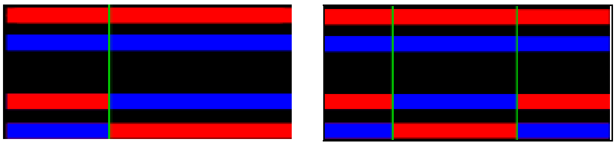
This step is used to generate the initial population. Initially many individual solutions are randomly generated to form an initial population. The population size depends on the nature of the problem, but typically contains several hundreds or thousands of possible solutions. Traditionally, the population is generated randomly, covering the entire range of possible solutions (the *search space*). Occasionally, the solutions may be "seeded" in areas where optimal solutions are likely to be found.

B. Selection

It is the stage of a genetic algorithm in which individual genomes are chosen from a population for later breeding (recombination or crossover). There are different approaches for making this selection. Two commonly used selection techniques are:

Roulette - A selection operator in which the chance of a chromosome getting selected is proportional to its fitness (or rank). This is where the concept of survival of the fittest comes into play.

Tournament - A selection operator which uses roulette selection N times to produce a tournament subset of chromosomes. The best chromosome in this subset is then chosen as the selected chromosome. This method of selection applies addition selective pressure over plain roulette selection.



(a) (b)

Figur1. (a) Single point crossover (b) Two point crossover

C. Crossover

Crossover is a genetic operator used to vary the programming of a chromosome or chromosomes from one generation to the next. It is analogous to reproduction and biological crossover, upon which genetic algorithms are based. However, here after the parents are selected, they undergo crossover only with a probability of P_{cross}. Many crossover techniques exist. Some of them are

1. Single point crossover

As shown in Fig.1 (a) single crossover point on both parents' organism strings is selected. All data beyond that point in either organism string is swapped between the two parent organisms. The resulting organisms are the children.

2. Two point crossover

As shown in Fig.1 (b), two-point crossover calls for two points to be selected on the parent organism strings. Everything between the two points is swapped between the parent organisms, rendering two child organisms.

D. Mutation

In genetic algorithms, mutation is a genetic operator used to maintain genetic diversity from one generation of a population of chromosomes to the next. It is analogous to biological mutation.

The classic example of a mutation operator involves a probability that an arbitrary bit in a genetic sequence will be changed from its original state. A common method of implementing the mutation operator involves generating a random variable for each bit in a sequence. This random variable tells whether or not a particular bit will be modified. The probability of mutation is given by P_{mut} which is usually at-least 100 times less than P_{cross}. The purpose of mutation in GAs is to allow the algorithm to avoid local minima by preventing the population of chromosomes from becoming too similar to each other, thus slowing or even stopping evolution. This reasoning also explains the fact that most GA systems avoid only taking the fittest of the population in generating the next but rather a random (or semi-random) selection with a weighting toward those that are fitter.

E. Termination

The generation process is repeated until a termination condition has been reached. Common terminating conditions are:

- A solution is found that satisfies minimum criteria.
- Fixed number of generations reached.
- Allocated budget (computation time/money) reached.

IV. PROPOSED SOLUTION

The following assumption had been made in the sub channel allocation algorithm $N_1 : N_2 : \ldots : N_K = \phi_1 : \phi_2 : \ldots : \phi_K$

The above assumption holds true and is a valid assumption as used in [8]. The proposed steps are as follows:

Step 1: Input number of generations (G), Channel gain matrix (H), Total Power (P_{tot}), Bandwidth (B), number of channels (N), number of users (K), and Proportional fairness ratio $\phi_1 : \phi_2 : \ldots : \phi_K$.

Step 2: Create initial population of chromosomes and calculate their capacities using $\frac{B}{N}\sum_{k=1}^{K}\sum_{n=1}^{N} c_{k,n} \log_2(1+p_{k,n}H_{k,n})$

Step 3: Sort all chromosomes in the increasing order of their capacities.

Step 4: Generate probability levels for each chromosome based on their total capacities that is, the chromosome with highest capacity has highest probability of selection for crossover.

Step 5: Selection of parent chromosomes that is select two chromosomes from the entire population.

Step 6: Select two points randomly on the parent chromosome for crossover.

Step 7: Generate two child chromosomes by swapping parent chromosomes from starting to first random point and from second random point to end of chromosome.

Step 8: Replace the two chromosomes of lowest capacities with the two new child chromosomes.

Step 9: Select a chromosome from the entire population for mutation.

Step 10: Exchange the channels allocated to users in the selected chromosome.

Step 11:If user has more channels than he should, corresponding channel is allocated to the user with lesser channels.

Step 12: If any user has less number of channels, unallocated channel with highest channel gain is allocated to that user.

Step 13: Calculate capacities for each chromosome using $\frac{B}{N}\sum_{k=1}^{K}\sum_{n=1}^{N} c_{k,n} \log_2(1+p_{k,n}H_{k,n})$

Step 14: Decrement G

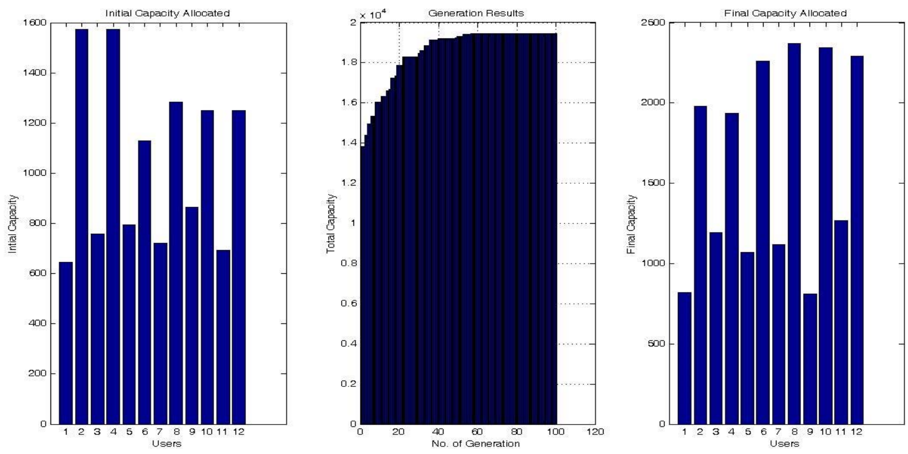
Step 15: If G = 0 select the chromosome with highest capacity, otherwise go to Step 4.

V. SIMULATION RESULTS

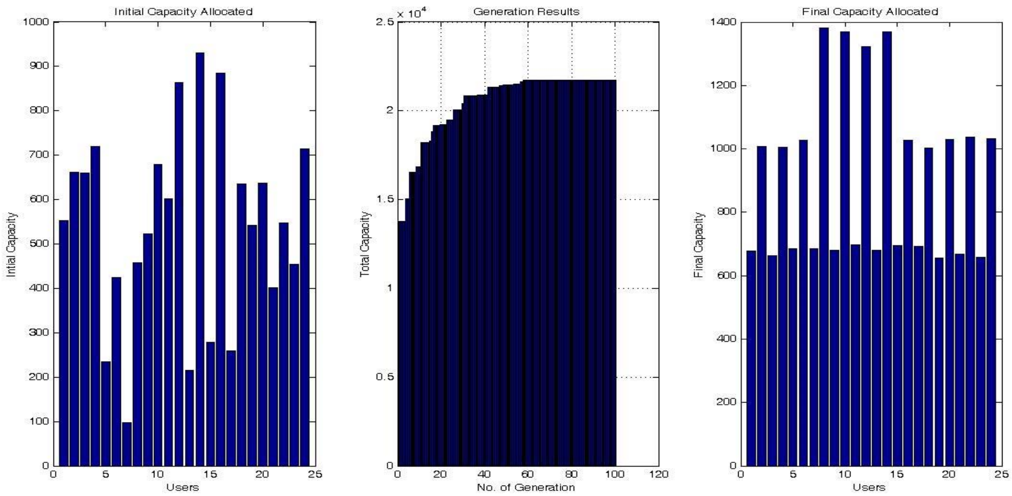
The frequency selective multipath channel had been modeled as consisting of six independent Rayleigh multipaths, with an exponentially decaying profile. A maximum delay spread of 5µs and maximum Doppler of 30 Hz were assumed. The channel information was sampled every 0.5 ms to update the subchannel and power allocation. The total power was assumed to be 1 W, the total bandwidth as 1 MHz, and total subcarriers as 64. The average subchannel SNR was taken as 38 dB, and BER $\leq 10^{-3}$, giving an SNR gap Γ= -ln (5 x 10^{-3})/1.6 = 3.3. This constant was used in the calculation of the rate $r_{k,n}$ of user k in subcarrier n given in (2).

The proportional rate constraint $\Phi_1 : \Phi_2 : \ldots : \Phi_k = 1:2:1:\ldots:2$ was taken. The number of users considered was 12, 16, 20 and 24. The results after performing the subchannel allocation are shown (for 12 and 24 users) in Fig.2. The three graphs from left to right show the initial capacity of each user, the final capacities after each generation and the final capacity of each user after the sub channel allocation.

As it can been be seen from the graphs, the algorithm presented is giving consistent good results for different number of users. After the above sub channel allocation is performed, the power allocation is done using standard water filling algorithm.



(a)



(b)
Figure 2. Simulation results of subcarrier allocation for number of users (a) 12 users (b) 24 users.

VI. CONCLUSION

This paper presents a new method to solve the rate-adaptive resource allocation problem with proportional rate constraint for OFDMA systems. It improves on the previous work in this area [8] by developing a novel GA for subcarrier allocation scheme that achieves approximate rate proportionality while maximizing the total capacity. This scheme was also able to exploit the special linear case in [14], thus allowing the optimal power allocation to be performed using a direct algorithm with a much lower complexity versus an iterative algorithm. It is shown through simulation that the proposed method performs better than the previous work in terms of achieving higher total capacities, while being applicable to a more general class of systems. However, the algorithm presented is slightly demanding computationally. The future scope of this paper is to try and bring down the computational demands of the algorithm presented.

REFERENCES

[1] E. Lawrey, .Multiuser OFDM,. in Proc. International Symposium on Signal Processing Applications'99, vol. 2, pp. 761.764, 1999.

[2] T. S. Rappaport, A. Annamalai, R. M. Beuhrer, and W. H. Tranter, Wireless Communications: Past Events and a Future Perspective. IEEE Commun. Mag., vol. 40, no. 5, pp. 148.161, May 2002.

[3] C. Y. Wong, R. S. Cheng, K. B. Lataief and R. D. Murch, "Multiuser OFDM System with Adaptive Subcarrier, Bit and Power Allocation" IEEE J. Select. Areas Commun, Vol. 17, pp. 1747 to 1758, October 1999.

[4] J. Jang and K. B. Lee, "Transmit Power Adaptation for Multiuser OFDM System" IEEE J. Select. Areas Commun, Vol. 21, pp. 171 to 178, February 2003.

[5] Y. B. Reddy, "Genetic Algorithm Approach for Adaptive Subcarrier, Bit, and Power Allocation" Proceedings of the 2007 IEEE International Conference in Networking, Sensing and Control, London, UK, 15-17 April 2007.

[6] Zhihua Tang, Youtuan Zhu, Guo Wei, Jinkang Zhu, "Cross-Layer Resource Allocation for Multiuser OFDM Systems based on ESGA" Proceedings of the 66th IEEE Vehicular Technology Conference (VTC 2007 – Fall), Baltimore, MD, USA, October 2007.

[7] Inhyoung Kim, In-Soon Park, and Yong H. Lee, "Use of Linear Programming for Dynamic Subcarrier and Bit Allocation in Multiuser OFDM" IEEE Transactions on Vehicular Technology, Vol. 55, No.4, July 2006.

[8] Ian C. Wong, Zukang Shen, Brian L. Evans and Jeffrey G. Andrews, "A Low Complexity Algorithm for Proportional Resource Allocation in OFDMA Systems" Proc. IEEE Int. Workshop on Signal Processing Systems, pp. 1-6, Oct. 2004.

[9] Nitin Sharma, Rao. A, Dewan. A, Safdari, M "Rate adaptive resource allocation for multiuser OFDM using NSGA – II" Proceeding of Fourth IEEE Conference on Wireless Communication and Sensor Networks (WCSN-2008), IIITA & DAVV, Indore, IEEE Catalog Number. CFP0875D, ISBN: 978-1-4244-3328-5, Library of Congress: 2008909475, Pp: 161-166, Dec. 2008,

[10] S. T. Chung and A. Goldsmith, "Degrees of Freedom in Adaptive Modulation: A Unified View", IEEE Trans. Commu., vol. 49, no. 9,pp. 1561.1571, Sept 2001.

[11] David A Coley, "An Introduction to Genetic Algorithms for Scientists and Engineers", World Scientific, ISBN:981-02-3602-6, 2003

[12] David E. Goldberg, "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison-Wesley, 1989.

[13] Melanie Mitchell, "An Introduction to Genetic Algorithms", MIT Press, ISBN: 0-262-13316-4, 1996.

[14] Z. Shen, J. G. Andrews, and B. L. Evans, .Optimal Power Allocation in Multiuser OFDM Systems,. in Proc. IEEE Global Communications Conference, San Francisco, CA, pp. 337.341, Dec. 2003.