PAPER

A Computationally Efficient Method for Large Dimension Subcarrier Assignment and Bit Allocation Problem of Multiuser OFDM System

Shin-Yeu LIN[†a)], ***Nonmember and*** **Jung-Shou HUANG**[†b)], ***Member***

SUMMARY In this paper, we propose a computationally efficient method to solve the large dimension Adaptive Subcarrier Assignment and Bit Allocation (ASABA) problem of multiuser orthogonal frequency division multiplexing system. Our algorithm consists of three Ordinal Optimization (OO) stages to find a good enough solution to the considered problem. First of all, we reformulate the considered problem to separate it into subcarrier assignment and bit allocation problem such that the objective function of a feasible subcarrier assignment pattern is the corresponding optimal bit allocation for minimizing the total consumed power. Then in the first stage, we develop an approximate objective function to evaluate the performance of a subcarrier assignment pattern and use a genetic algorithm to search through the huge solution space and select s best subcarrier assignment patterns based on the approximate objective values. In the second stage, we employ an off-line trained artificial neural network to estimate the objective values of the s subcarrier assignment patterns obtained in stage 1 and select the l best patterns. In the third stage, we use the exact objective function to evaluate the l subcarrier assignment patterns obtained in stage 2, and the best one associated with the corresponding optimal bit allocation is the good enough solution that we seek. We apply our algorithm to numerous cases of large-dimension ASABA problems and compare the results with those obtained by four existing algorithms. The test results show that our algorithm is the best in both aspects of solution quality and computational efficiency.
key words: *OFDM system, combinatorial optimization, ordinal optimization, resource allocation, wireless communication*

1. Introduction

The Adaptive Subcarrier Assignment and Bit Allocation (ASABA) for multiuser Orthogonal Frequency Division Multiplexing (OFDM) system has been studied for a number of years. This issue is initialized by Wong et al. in [1] and is formulated as a nonlinear integer programming problem to minimize the total power consumption while satisfying the users' data communication request and system's constraints. Since then, various heuristic methods, ranging from the more computation-time consuming and global-like mathematical programming based approaches [1], [2] to the less computation-time consuming and more local-like two module scheme [3], two-step subcarrier assignment approaches [4], [5] and iterative grouping scheme [6], were proposed to cope with this NP-hard *constrained combinatorial* optimization problem.

In recent years, due to the considerable increase in the number of mobile users in wireless network, and the expanding capacity of the multiuser OFDM system to meet the increasing wireless communication demand, the dimension of the ASABA problem is growing. The increasing dimension will (i) adverse the computational complexity of the already time consuming mathematical programming based approaches [1], [2] and (ii) enlarge the discrete solution space, which will degrade the quality of the solutions obtained by the more local-like approaches [3]–[6] as well as the corresponding computation time. Thus, dealing with large-dimension ASABA problem is a challenging issue in wireless communication, and the *purpose* of this paper is proposing a computationally efficient method to solve the considered problem for a *good enough* solution. The quality of the solution obtained by the mathematical programming based approach [1] is considered to be one of the best so far. However, they arbitrarily round the optimal *continuous* subcarrier assignment pattern off to the closest discrete values may cause *infeasibility* problem and *not* guarantee to be a good solution, if feasible. To avoid the undesirable effect caused by rounding off, we will handle the discrete solution space directly and use a global-like approach. However, the global searching techniques [7] such as *Genetic Algorithm* (GA), *Simulated Annealing* method, *Tabu Search* method and *Evolutionary Programming* are not adequate here because of their tremendous computation time, which is even worse than the mathematical programming based approach due to (i) evaluating the *objective value* of a feasible subcarrier assignment pattern is time consuming, (ii) handling the constraints is not an easy task and (iii) the size of the discrete solution space is huge. Evaluating the exact performance (i.e. the objective value) of a feasible subcarrier assignment pattern is a conventional "*value*" concept. However, it is indicated in a recently developed optimization technique, the *Ordinal Optimization* (OO) theory [8], [9], that the performance order of discrete solutions is likely preserved even evaluated by a *surrogate model*. In other words, the OO theory claims that there is high probability that we can find the *actual* good discrete solutions if we limit ourselves to the top $n\%$ of the estimated good discrete solutions evaluated by a surrogate model [10]. Thus, to retain the merit of global searching technique while avoiding the cumbersome

Manuscript received January 17, 2008.
Manuscript revised May 6, 2008.
[†]The authors are with the Department of Electrical and Control Engineering, National Chiao-Tung University, Hsinchu, 300, Taiwan, R.O.C.
a) E-mail: sylin@cc.nctu.edu.tw
b) E-mail: rong@emc.com.tw
DOI: 10.1093/ietcom/e91–b.12.3966

conventional performance evaluation of a discrete solution, our approach is based on OO theory to solve the considered problem for a *good enough* solution with *high probability* using *limited* computation time.

Our approach consists of three OO stages. First of all, we will reformulate the considered problem to separate it into subcarrier assignment and bit allocation problem such that the objective function of a feasible subcarrier assignment pattern is the corresponding optimal bit allocation for minimizing the total consumed power. Then, in the first stage, we will develop an *easy-to-evaluate approximate* objective function to *estimate* the objective value of a subcarrier assignment pattern and employ a GA to search through the huge discrete solution space to find the top s subcarrier assignment patterns based on the estimated objective values. In the meantime, a subtle *representation scheme* and a *repair operator* for GA need be designed to handle the constraints of the considered problem. In the second stage, we use an off-line trained *Artificial Neural Network* (ANN) to estimate the objective values of the s subcarrier assignment patterns obtained in stage 1 and pick the top l patterns based on the estimated objective value. In the third stage, we use the exact objective function to evaluate the l subcarrier assignment patterns obtained in stage 2, and the best one associated with the corresponding optimal bit allocation is the good enough solution that we seek. In the proposed three-stage approach, the models employed to evaluate a solution are varying from very rough (stage 1) to exact (stage 3). In the meantime, the candidate solution space is reduced from the *original* huge solution space (stage 1) to only *l candidate solutions* (stage 3). In general, a more accurate approximate objective function will take more time to evaluate a solution; however as can be seen from our three-stage approach, when a more accurate approximate objective function is used, the search space is already reduced considerably, and the computation time is largely reduced accordingly.

Our paper is organized in the following manner. In Sect. 2, we will state the considered problem and its reformulation. In Sect. 3, we will present our three OO stages to solve the considered problem. In Sect. 4, we will apply our algorithm to numerous large-dimension ASABA cases and compare with some existing algorithms in the aspects of solution quality and computation time. Finally, we will draw a conclusion in Sect. 5.

2. Problem Statement and Reformulation

2.1 Problem Statement

As shown in Fig. 1, we assume that the system has K users to share N subcarriers. Each user's data rate request will be allocated to a nonoverlapping set of subcarriers and distributed among them. It is also assumed that a subcarrier cannot be shared by more than one user.

In the transmitter part of Fig. 1, the serial data from K users are fed into the block represented by the proposed ASABA algorithm. The algorithm will be executed in every allocating period to assign the set of subcarriers to each user and the number of bits to be transmitted on each assigned subcarrier based on the updated channel information of all users. Then proper modulation scheme and transformation scheme will be used to transform the assigned bits into time domain samples, which will be transmitted to the receivers by different frequency selective fading channels to different users as shown in Fig. 1.

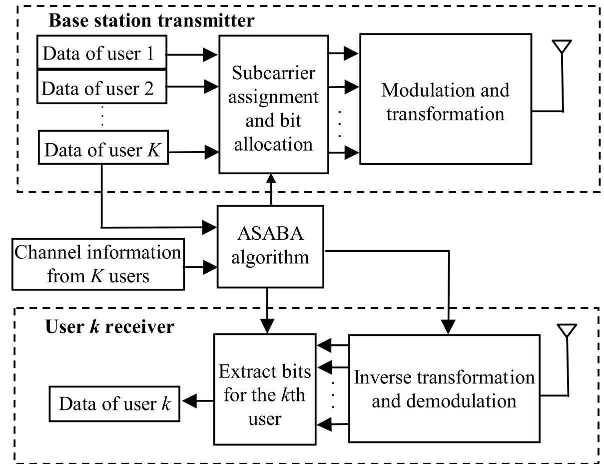

Fig. 1 Block diagram of a multiuser OFDM system with subcarrier assignment and bit allocation.

We assume the subcarrier assignment and bit allocation information is sent to the receivers via a separate control channel. For the sake of simplicity, we only show the receiver part of one user in Fig. 1. The received time domain samples of the kth user is transformed back and demodulated to bits from the subcarriers assigned to the kth user based on the subcarrier assignment and bit allocation information.

In this paper, we focus on proposing a computationally efficient method, the ASABA algorithm, to solve the following ASABA problem of the multiuser OFDM system[†] for a good enough feasible solution.

$$\min_{c_{k,n},\rho_{k,n}} P_T \left(= \sum_{n=1}^{N} \sum_{k=1}^{K} \frac{\rho_{k,n}}{\alpha_{k,n}^2} f_k(c_{k,n}) \right)$$
$$\text{subject to } 0 \le c_{k,n} \le M, \text{ for all } k, n,$$
$$R_k = \sum_{n=1}^{N} \rho_{k,n} c_{k,n}, \; k = 1, \ldots, K,$$
$$\sum_{k=1}^{K} \rho_{k,n} = 1, \; n = 1, \ldots, N,$$
$$\rho_{k,n} \in \{0, 1\}, \text{ for all } k \text{ and } n \tag{1}$$

where K denotes the number of users in the system that con-

[†]There are two types of formulations on this issue. One is the Margin Adaptive (MA) optimization, whose formulation is (1), and the other is the Rate Adaptive (RA) optimization, which maximizes the data rate under a power constraint [2]. Kim et al. had shown in [2] that the RA optimization problem can be solved via recursive MA optimization.

sists of N subcarriers for data transmission; $\alpha_{k,n}$ is the channel gain for user k using subcarrier n, and $\frac{f_k(c_{k,n})}{\alpha_{k,n}^2}$ denotes the required transmission power for user k to use subcarrier n; P_T denotes the total transmission power to be minimized; $\rho_{k,n} \in \{0, 1\}$ is an indicator variable such that $\rho_{k,n} = 1$ means subcarrier n is assigned to user k, and 0 otherwise; in the meantime, a subcarrier can be occupied by at most one user as described by the second equality constraint; R_k (bits per OFDM *symbol*) denotes the requested data rate of the kth user; $c_{k,n}$, an integer, denotes the number of bits of the kth user assigned to the nth subcarrier, and M denotes the maximum available number of bits in a subcarrier; thus the first equality constraint implies that the subcarrier assignment and bit allocation should meet the user's data rate request. Clearly, (1) is a nonlinear integer programming problem or a constrained combinatorial optimization problem, because $\rho_{k,n}$ and $c_{k,n}$ are integers for all k, n, and P_T is nonlinear.

We *assume* the solution of (1) exists, which is equivalent to the following *assumption*: there are enough subcarriers to meet the data rate request of all users, i.e. the following inequality hold

$$\sum_{k=1}^{K} \left\lceil \frac{R_k}{M} \right\rceil \leq N \tag{2}$$

because a subcarrier cannot be shared by more than one user. Notation $\lceil y \rceil$ in (2) denotes the integer closest to y on the right-hand side.

Remark 1: (i) The situation that the solution of (1) does not exist, i.e. assumption (2) does not hold is beyond the scope of this paper. (ii) For the extreme case that $\sum_{k=1}^{K} \left\lceil \frac{R_k}{M} \right\rceil = N$ and the spectrum is *fixed* for each user, (1) becomes a very simple optimal bit allocation problem and can be readily solved by the existing greedy algorithm [11]. However, methods proposed in this paper and [1]–[4] aim to solve ASABA problem, (1), which is much more complicated than the above mentioned extreme case.

2.2 Reformulation

To develop an approximate objective function for a subcarrier assignment pattern, we need to reformulate (1). We let ρ, c, α and R denote the vectors $[\rho_{k,n}]$, $[c_{k,n}]$, $[\alpha_{k,n}]$, $k = 1, \ldots, K, n = 1, \ldots, N$, and $[R_k]$, $k = 1, \ldots, K$, respectively, and define $C(\rho)$, *the feasible set of bit allocation c for a given* ρ, as

$$C(\rho) = \left\{ c | 0 \leq c_{k,n} \leq M \text{ and } R_k = \sum_{n=1}^{N} \rho_{k,n} c_{k,n}, \right.$$
$$\left. k = 1, \ldots, K, n = 1, \ldots, N, \text{ for a given } \rho \right\} \tag{3}$$

Now for a given subcarrier assignment pattern ρ that satisfies $\sum_{k=1}^{K} \rho_{k,n} = 1$, $n = 1, \ldots, N$ and $\rho_{k,n} \in \{0, 1\}$ for all k and n, (1) becomes an *optimal bit allocation problem* under the given ρ that is to find the optimal c of the following problem:

$$\min_{c} \sum_{n=1}^{N} \sum_{k=1}^{K} \frac{\rho_{k,n}}{\alpha_{k,n}^2} f_k(c_{k,n})$$
$$\text{subject to } c \in C(\rho)$$
$$C(\rho) \neq \phi \tag{4}$$

where the constraint $C(\rho) \neq \phi$ represents the existence of feasible bit allocation c for the given ρ. Note that the assumption, (2), of our problem does not imply $C(\rho) \neq \phi$ for any ρ that satisfies $\sum_{k=1}^{K} \rho_{k,n} = 1$, $n = 1, \ldots, N$ and $\rho_{k,n} \in \{0, 1\}$; for example, in an extreme case that $\rho_{1,1} = \rho_{1,2} = \ldots = \rho_{1,N} = 1$ and the rest of $\rho_{k,n} = 0$, then if (2) is satisfied but $R_k \neq 0$ for any $k \neq 1$, we have $C(\rho) = \phi$ for the given ρ. However, the assumption (2) guarantees that there exists ρ such that $C(\rho) \neq \phi$. In fact, the constraint $C(\rho) \neq \phi$ is equivalent to the inequality constraints $R_k \leq M \sum_{n=1}^{N} \rho_{k,n}$, $k = 1, \ldots, K$. Thus, we can rewrite (1) into the following form:

$$\min_{\rho} \left\{ \min_{c} \sum_{n=1}^{N} \sum_{k=1}^{K} \frac{\rho_{k,n}}{\alpha_{k,n}^2} f_k(c_{k,n}) | c \in C(\rho) \right\}$$
$$\text{subject to } R_k \leq M \sum_{n=1}^{N} \rho_{k,n}, \quad k = 1, \ldots, K,$$
$$\sum_{k=1}^{K} \rho_{k,n} = 1, \ n = 1, \ldots, N,$$
$$\rho_{k,n} \in \{0, 1\}, \text{for all } k \text{ and } n \tag{5}$$

(5) can be viewed as a *separation* of *subcarrier assignment* and *bit allocation* problem, because the optimization problem inside the big bracket is the *optimal bit allocation* problem for a given *feasible* ρ that satisfies all the constraints in (5), and the overall problem is finding the best feasible ρ associated with an optimal bit allocation. Furthermore, the optimal bit allocation problem is separable for a given feasible ρ, because its objective function $\sum_{n=1}^{N} \sum_{k=1}^{K} \frac{\rho_{k,n}}{\alpha_{k,n}^2} f_k(c_{k,n})$, as well as the constraints $c \in C(\rho)$ are separable. Thus we can decompose it into the following K independent subproblems: For $k = 1, \ldots, K$,

$$\min_{c_{k,1}, \ldots, c_{k,N}} \sum_{n=1}^{N} \frac{\rho_{k,n}}{\alpha_{k,n}^2} f_k(c_{k,n})$$
$$\text{subject to } 0 \leq c_{k,n} \leq M, n = 1, \ldots, N,$$
$$R_k = \sum_{n=1}^{N} c_{k,n} \tag{6}$$

Clearly (5) is a constrained combinatorial optimization problem with (i) hard to evaluate objective function, because the objective function $\{\min \sum_{n=1}^{N} \sum_{k=1}^{K} \frac{\rho_{k,n}}{\alpha_{k,n}^2} f_k(c_{k,n}) | c \in C(\rho)\}$ itself is an optimization problem, (ii) equality and inequality constraints involving integers and (iii) huge discrete solution space for ρ as long as K and N are large.

Remark 2: As indicated previously, the assumption (2) is to assume that the solution of (1) exists. Since (5) is equivalent to (1), the assumption (2) should also apply to (5) to assume the solution of (5) exists. It should be noted that no additional assumption is needed to derive (5) from (1). Furthermore, the decomposition of the objective function of (5), i.e. the term inside the big bracket of (5), into (6) has nothing to do with the assumption (2). This decomposition simply says that for a given feasible ρ, the optimal bit allocation for individual user is independent of each other.

3. The Three-Stage Ordinal Optimization (OO) Approach — The ASABA Algorithm

The proposed ASABA algorithm for solving (5), or (1) equivalently, consists of three OO stages as stated in the following.

3.1 Stage 1: Using GA to Select Top s Subcarrier Assignment Patterns Based on an Easy-to-Evaluate Approximate Objective Function

As shown in (6) that the objective function of (5) for a given feasible ρ can be decomposed into K independent optimal bit allocation subproblems. We let $N_k = \sum_{n=1}^{N} \rho_{k,n}$ and $\hat{\alpha}_k = \frac{\sum_{n=1}^{N} \rho_{k,n}\alpha_{k,n}}{N_k}$ denote the total number of subcarriers assigned to and the average power consumption coefficient of user k, respectively. Then, we use the following to *approximate* the optimal power consumed by user k, i.e. the optimal objective value of (6), for the given ρ. We assume the total data rate request R_k of user k are distributed *equally* to the assigned N_k subcarriers, i.e. setting $c_{k,n} = \frac{R_k}{N_k}$ for each assigned subcarrier, and assume each of the N_k subcarriers has the same power consumption coefficient $\hat{\alpha}_k$ defined above, then the power consumed by user k, denoted by $\hat{P}_k$, can be computed by $\hat{P}_k = \frac{N_k}{\hat{\alpha}_k^2} f_k(\frac{R_k}{N_k})$. Consequently, we can obtain the *approximate* total consumed power for the given ρ, denoted by $\hat{P}_T$, by calculating $\hat{P}_T = \sum_{k=1}^{K} \hat{P}_k$, which will serve as the *approximate* objective function of (5).

Then, to use GA as a global searching technique, we need to define a *representation scheme* to map all ρ that satisfy $\rho_{k,n} \in \{0, 1\}$ and $\sum_{k=1}^{K} \rho_{k,n} = 1$ into a set of *chromosomes* first [7], [12, Ch.14]. Let the *alphabet* of the representation scheme be the set $\{1, 2, \ldots, K\}$. We define the chromosome u as a string of N symbols, $u_1, u_2, \ldots, u_N$, such that the nth symbol u_n, which takes an element from the alphabet, indicates the user that subcarrier n is assigned to. In other words, $u_n = k_1$ means $\rho_{k_1,n} = 1$ and $\rho_{k,n} = 0$ for all $k \neq k_1$. This representation scheme ensures that $\rho_{k,n} \in \{0, 1\}$ and satisfies $\sum_{k=1}^{K} \rho_{k,n} = 1$, because u_n taking only one element from the alphabet implies that the nth subcarrier can at most be assigned to one user. However not all chromosomes u can satisfy the inequality constraints $M \sum_{n=1}^{N} \rho_{k,n} \geq R_k$ required in (5). The required number of subcarriers for user k to meet the inequality constraint is at least $\left\lceil \frac{R_k}{M} \right\rceil$. We define $\delta_k(u_n) = 1$ if $u_n = k$ and 0, otherwise. Thus, the number of subcarriers assigned to user k in a chromosome u is $\sum_{n=1}^{N} \delta_k(u_n)$. To meet the inequality constraint, the following has to hold

$$\sum_{n=1}^{N} \delta_k(u_n) \geq \left\lceil \frac{R_k}{M} \right\rceil \quad (7)$$

We define $\sigma_k(u) = \sum_{n=1}^{N} \delta_k(u_n) - \left\lceil \frac{R_k}{M} \right\rceil$, then $\sigma_k(u) \geq 0$ implies the number of subcarriers assigned to user k is enough or surplus, and $\sigma_k(u) < 0$ implies the other way. Therefore, for a given u we can compute $\sigma_k(u)$ for $k = 1, \ldots, K$ and order them in an ascending sequence $\sigma_{k_1}(u) \leq \sigma_{k_2}(u) \leq \ldots \leq \sigma_{k_K}(u)$, where the ordered indices $k_1, \ldots, k_K \in \{1, \ldots, K\}$, and we have no order preference for the k's with same values of $\sigma_k(u)$. Thus $\sigma_{k_1}(u) \geq 0$ implies u satisfies (7) for all k and is feasible.

Suppose $\sigma_{k_1}(u) < 0$, then u is infeasible. Such an *infeasibility* problem may occur to any newly generated chromosomes, resulted from *initial population* generation, *crossover* operations, and *mutation* operations, and can be resolved by a *repair operator*, which is designed to recover the infeasible chromosome to a feasible one as stated in the following. Under the *assumption* that we have enough subcarriers to meet all users' data rate request, (2), if $\sigma_{k_1}(u) < 0$, there must exist i such that $\sigma_{k_1}(u) \leq \ldots \leq \sigma_{k_i}(u) < 0$, $\sigma_{k_K}(u) \geq \ldots \geq \sigma_{k_{i+1}}(u) \geq 0$, and $\sigma_{k_{i+1}}(u) + \ldots + \sigma_{k_K}(u) \geq -(\sigma_{k_1}(u) + \ldots + \sigma_{k_i}(u))$. Thus, we can reassign the surplus subcarriers of users $k_{i+1}, \ldots, k_K$ to users $k_1, \ldots, k_i$ in the following manner. Randomly pick the surplus subcarriers that were assigned to user k_K to make up the insufficient subcarriers required by k_1, that is randomly pick an u_m from all u_m's with $u_m = k_K$, and reset the picked u_m as $u_m = k_1$. When all surplus subcarriers of user k_K are reassigned, we proceed with picking the surplus subcarriers of user k_{K-1} and so forth. Similarly, when the number of insufficient subcarriers of user k_1 is made up, we proceed with making up the insufficient subcarriers of user k_2 and so forth. The above process will continue until the number of insufficient subcarriers of user k_i is made up. Then the resulting u will be feasible. Based on this repair operator, we may describe the employed GA to solve (5) in the following.

We randomly generate I, say 200, chromosomes such that each symbol of each chromosome is assigned by an element randomly selected from the alphabet, $\{1, \ldots, K\}$, and apply the repair operator to them. The resulting I feasible chromosomes will serve as the initial population of the employed GA. To evaluate the *fitness* of a chromosome u based on the above mentioned *approximate* total power consumption, we first compute $N_k(u) = \sum_{n=1}^{N} \delta_k(u_n)$, $\hat{\alpha}_k(u) = \frac{\sum_{n=1}^{N} \delta_k(u_n)\alpha_{k,n}}{N_k(u)}$, and $\hat{P}_k(u) = \frac{N_k(u)}{\hat{\alpha}_k^2} f_k(\frac{R_k}{N_k(u)})$ for every $k = 1, \ldots, K$. Then, the fitness of u will be $\frac{1}{\hat{P}_T(u)}$, where $\hat{P}_T(u) = \sum_{k=1}^{K} \hat{P}_k(u)$. Based on the fitness values of all chromosomes in the *population* pool, we use *roulette wheel selection scheme* to select chromosomes into the *mating* pool, from which we select chromosomes to serve as the *parents* for crossover. The probability that a chromosome is selected as a parent is p_c, say 0.7.

We apply a *single point crossover scheme* to the selected parents, and the generated *offsprings* may be infeasible as indicated previously. Therefore we will apply the repair operator to each generated offspring, and the resulting feasible offsprings will replace the corresponding parents in the mating pool. Subsequently, we will apply the *mutation* operation to each chromosome in the mating pool with *mutation probability* p_m, say 0.02. Any changed chromosome after mutation operation may also be infeasible, and we will apply the repair operator to it. Consequently, the resulting chromosomes in the mating pool after the above *evolution process* will be the population pool of next iteration.

The above process completes one iteration of our GA. We stop the GA when the number of iterations exceeds 60. After the applied GA converges, we rank the final I chromosomes (i.e. u's) based on their fitness values and pick the best s (=50) u's. We can then convert these s chromosomes into the subcarrier assignment patterns ρ's in the following manner: $\rho_{k,n} = \delta_k(u_n)$ for all k and n. Then these converted ρ's are the s subcarrier assignment patterns determined in this stage, and they are feasible for (5).

Remark 3: Based on [10], larger s will consist of more actual good subcarrier assignment patterns. However, larger s may cause more computation time for further evaluation. Thus, the value of s should be determined based on the available computation budget to obtain and the required goodness of the good enough solution. In the current application, the computation time is of more concern.

3.2 Stage 2: Choose Top l Subcarrier Assignment Patterns from the s Based on an ANN Model

Since evaluating the s ρ's obtained in Stage 1 using the exact objective function is still too time consuming, based on [8], we can trim the candidate solution set further using a more accurate approximate objective function. Therefore, we will employ a *supervised learning* ANN [13] to estimate the optimal power consumed by user k and select top l (=3) ρ's from the s.

Remark 4: The value of l is also determined based on a tradeoff between the computation time required to obtain and the goodness of the good enough solution.

This ANN is trained *off-line* using 5000 input/output pairs of data. The input data associated with user k is the data rate request R_k, the number of subcarriers assigned to user k, N_k, and the power consumption coefficient α_{k,n_i}, $i = 1, \ldots, N_k$, for the assigned N_k subcarriers $n_1, \ldots, n_{N_k}$. However, the dimension of the vector $(R_k, \alpha_{k,n_1}, \ldots, \alpha_{k,n_{N_k}})$ is large provided that N_k is large. A large ANN, i.e. an ANN consisting of large number of neurons in both input and hidden layers, will consume more computation time to obtain the output even if it is trained off-line. Although larger ANN can serve as a more accurate function approximator, what we care here is the performance order rather than the performance value. Therefore, for computation-time concern, we favor a simpler ANN. Since the values of $\alpha_{k,n_1}, \ldots, \alpha_{k,n_{N_k}}$ may have some kind of distribution, to characterize these values without using the details, we may use the corresponding mean, $\hat{\alpha}_k$, and variance, $\mathrm{var}(\alpha_k)$ [14]. Thus, to design a simple ANN, we will employ (R_k, N_k, $\hat{\alpha}_k$, $\mathrm{var}(\alpha_k)$) to characterize the input data of user k.

Consequently, the 5000 input/output pairs used to train the ANN can be obtained as follows. We uniformly select 5000 sets of (R_k, N_k) from the following ranges: $R_k \in [5, 150]$, $N_k \in [\frac{R_k}{M}, \frac{2R_k}{M}]$, which makes $R_k \leq MN_k$ to ensure that there are enough subcarriers to meet the data rate request. For each N_k, we randomly select $n_1, \ldots, n_{N_k}$ from $\{1, \ldots, N\}$, then randomly generate α_{k,n_i} from the range [0,2.0] for each $i = 1, \ldots, N_k$. The $\hat{\alpha}_k$ and $\mathrm{var}(\alpha_k)$ can be computed accordingly. Thus, we have 5000 input vectors of (R_k, N_k, $\hat{\alpha}_k$, $\mathrm{var}(\alpha_k)$). Now for each $(R_k, N_k, \alpha_{k,n_1}, \ldots, \alpha_{k,n_{N_k}})$, the corresponding output data for training ANN is the *actual* optimal power consumed by user k denoted by P_k. To compute P_k, we can use the greedy algorithm [11] to solve (6) by optimally distributing R_k bits to the assigned N_k subcarriers one at a time based on the *least incremental power consumption criteria*. We then use the obtained 5000 input/output pairs, ((R_k, N_k, $\hat{\alpha}_k$, $\mathrm{var}(\alpha_k)$), P_k), of data to train a three-layer ANN whose structure is described in the following. The input layer consists of four neurons corresponding to R_k, N_k, $\hat{\alpha}_k$, and $\mathrm{var}(\alpha_k)$. The hidden layer consists of 15 neurons, and each neuron uses *tansig* as the activation function. There is only 1 neuron in the output layer corresponding to P_k, and we use *purelin* as the activation function. Using the 5000 input/output pairs of data, we train this ANN by adjusting its arc weights using the Levenberg-Marquardt method proposed in [15], [16]. Based on this off-line trained ANN, we can estimate the total consumed power corresponding to a ρ as follows. For a given ρ, we can compute $N_k = \sum_{n=1}^{N} \rho_{k,n}$ and determine $\alpha_{k,n_1}, \ldots, \alpha_{k,n_{N_k}}$ from the gain α for each $k = 1, \ldots, K$. Subsequently, we can set up the input (R_k, N_k, $\hat{\alpha}_k$, $\mathrm{var}(\alpha_k)$), feed into the off-line trained ANN, and obtain the *estimated* P_k, denoted by $\tilde{P}_k$, from the output of ANN for each $k = 1, \ldots, K$. Then we can compute the estimated total consumed power, denoted by $\tilde{P}_T$, for the given ρ by $\tilde{P}_T = \sum_{k=1}^{K} \tilde{P}_k$. Using this off-line trained ANN, the l (=3) ρ's with smallest $\tilde{P}_T$ among the s feasible ρ's obtained in Stage 1 are the subcarrier assignment patterns determined in this stage.

3.3 Stage 3: Determine the Good Enough Subcarrier Assignment and Bit Allocation

Since there are only l (=3) candidate feasible-ρ's left, we can use the exact objective function of (5) to calculate the objective value of each ρ. That is to solve the optimal bit allocation problem (6) for the given ρ using the greedy algorithm mentioned above to obtain the optimal power consumption P_k for user $k = 1, \ldots, K$. Then we calculate $P_T = \sum_{k=1}^{K} P_k$ for the given ρ. Consequently, the ρ associated with the optimal bit allocation corresponding to the smallest P_T among the l feasible ρ's will be the good enough solution of (1) that we look for.

4. Test Results and Comparisons

In this section, we will demonstrate the performance of the proposed ASABA algorithm on solving the large-dimension ASABA problem (5), which is equivalent to (1), in the aspects of solution quality and computational efficiency by comparing with other algorithms. We assume the OFDM system has 256 subcarriers (i.e. $N = 256$), which can carry two, four, and six bits/symbol; therefore in this system $M = 6$. We adopt the following approximate formula employed in [1]–[4] for the $f_k(c)$ in the transmission power $\frac{f_k(c)}{\alpha_{k,n}^2}$ shown in the objective function of (1):

$$f_k(c) = \frac{N_0}{3}\left[Q^{-1}\left(\frac{P_e}{4}\right)\right]^2 (2^c - 1) \qquad (8)$$

where $Q^{-1}(x)$ is the inverse function of

$$Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^{\infty} e^{\frac{-t^2}{2}}\,dt \qquad (9)$$

P_e denotes Bit Error Rate (BER) and N_0 denotes the noise Power Spectral Density (PSD) level, and we set $P_e = 10^{-4}$ and $N_0 = 10^{-12}$ watt in the following simulations.

We use a frequency-selective channel consisting of six independent Rayleigh multipaths to model the wireless transmission channel, and each multipath is modeled by Clark's flat fading model [17]. We assumed that the power delay profile is exponentially decaying with e^{-2p}, where $p = 0, 1, 2, 3, 4$ and 5 denote the multipath index. Hence, the related power of the six multipath components are 0 dB, −8.69 dB, −17.37 dB, −26.06 dB, −34.74 dB, and −43.43 dB. We also assume the average subcarrier channel gain $E\left|\alpha_{k,n}\right|^2$ is unity for all k and n. Based on the above assumptions, we can generate power consumption coefficients $\alpha_{k,n}$, $k = 1, \ldots, K$, $n = 1, \ldots, N$, using MATLAB for our simulations.

We consider cases of various number of users for K=10, 20, 30, 40, and 50. For each K, we assume a fixed total data rate request R_T=1024 bits/symbol and randomly generate R_k, $k = 1, \ldots, K$, based on the constraint $\sum_{k=1}^{K} R_k = R_T$. For each K and the associated R, we randomly generate 5000 sets of $\alpha_{k,n}$, $k = 1, \ldots, K$, $n = 1, \ldots, N$, based on the above mentioned power consumption coefficient generation process and denote α^i as the ith set in the 5000. With the above test setup, we apply our algorithm to solve (1) on a Pentium 2.4 GHz processor and 512 Mbytes RAM PC. We also apply the more global-like mathematical programming based approaches proposed by Wong et al. and Kim et al. in [1] and [2], respectively, and the more local-like two-module scheme and two-step subcarrier assignment approaches proposed by Ergen et al. and Zhang in [3] and [4], respectively, to the same test cases on the same PC. For the purpose of comparison, we can use the average bit Signal-to-Noise Ratio (*abSNR*) to replace P_T, because *abSNR* is defined as the ratio of the average transmit power, $\frac{P_T}{R_T}$, to the noise PSD level N_0. As we have assumed that all the data rates per symbol are fixed at R_T, and the N_0 is just a constant, thus P_T is proportional to *abSNR*.

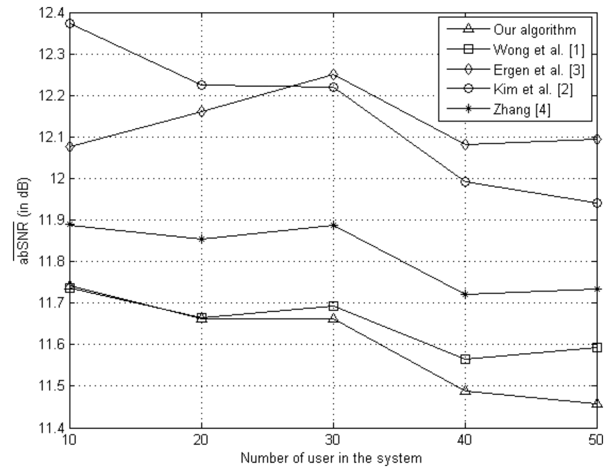

Fig. 2 The $\overline{abSNR}$ for K =10, 20, 30, 40 and 50 obtained by the five algorithms.

Remark 5: As shown in (8), P_T consists of the term N_0. Therefore, the magnitude of N_0 employed in our tests is not relevant to the results of *abSNR*, because the term N_0 will be cancelled out as noted in the definition of *abSNR*.

For each K with the associated vector R, we denote $abSNR(\alpha^i)$ as the resulted *abSNR* when α^i is used and calculate $\overline{abSNR} = \frac{\sum_{i=1}^{5000} abSNR(\alpha^i)}{5000}$, where $\overline{abSNR}$ denotes the average of the 5000 *abSNR*'s for a given K. The resulted $\overline{abSNR}$ for each K and each algorithm are shown in Fig. 2.

Form Fig. 2, we see that the $\overline{abSNR}$ obtained by our algorithm, which are marked by "△," is smallest among all algorithms. Moreover, the result obtained by our algorithm is even better when the number of users increases as can be observed from Fig. 2.

Remark 6: The quality of the solution obtained by the approach proposed by Wong et al. in [1] is excellent and has been used as a comparing standard in most of the literature regarding ASABA problems [2], [4], [5]. We also manifest the quality of their solution in our simulations as shown in Fig. 2. The reason that supports their solution's excellent quality is their global-like mathematical programming based approach as indicated in Sect. 1. They first employed a Lagrangian relaxation method to solve the continuous version of the ASABA problem then rounded the optimal continuous subcarrier assignment solution off to the closest integer solution. Such an arbitrarily rounding off may cause possible *infeasibility* and not *theoretically* guarantee to obtain a good solution, especially when the dimension of the ASABA problem is large. Dislike their approach, we handle the discrete solution space directly. In the first stage of our approach, our specially designed GA, which associates with a surrogate model for fast fitness evaluation, search through the whole feasible solution space to find some good feasible subcarrier assignment patterns. Thus, our approach is also global-like and will not cause any infeasibility problem. Then in the second and third stages, we use the ANN and

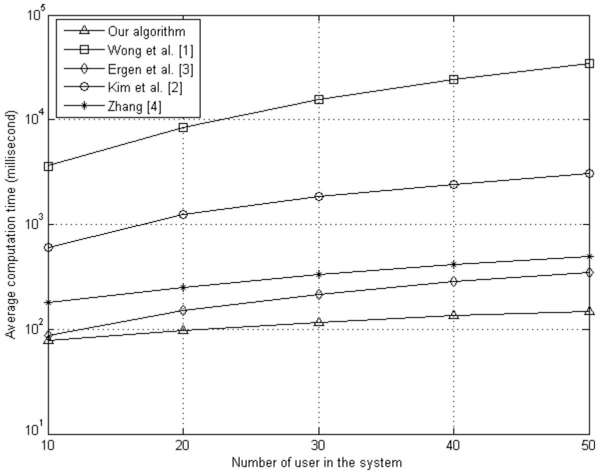

Fig. 3 The average computation time for obtaining an *abSNR* by the five algorithms in cases of K =10, 20, 30, 40 and 50.

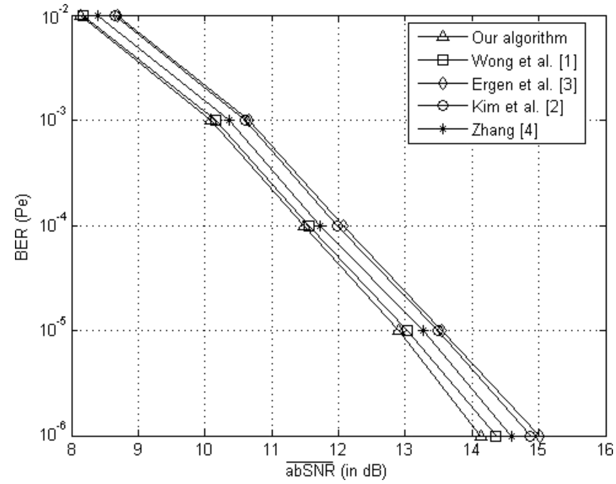

Fig. 4 Comparison of the five algorithms for various P_e in the case of K=40.

exact models, respectively, to help pinpoint a good enough subcarrier assignment pattern associated with optimal bit allocation among the feasible solutions resulted in Stage 1. The arbitrarily rounding off technique employed in [1] is lack of theoretical support. However, the foundation of our approach is OO theory, which is a theoretically sound general methodology [8] and has several successful applications on the combinational optimization problems with huge discrete solution space [18]–[20].

We also show the average computation time for obtaining an *abSNR* for each K and each algorithm in Fig. 3. From this figure, we see that the average computation time obtained by our algorithm, which are around 100 milliseconds as marked by "△," is also smallest among all algorithms. These results show that our algorithm outperforms the other four in both aspects of solution quality and computational efficiency. More importantly, when the number of users increases, the performance of our algorithm is even better. This demonstrates that our algorithm is most suitable for large-dimension ASABA problems.

Remark 7: The methods in [2]–[4] are proposed to overcome the computational complexity of the method in [1]. Indeed, the methods in [3], [4] are more computationally efficient than the methods in [1], [2] as shown in Fig. 3, because the former are local-like heuristic methods while the latter are global-like mathematical programming based approaches. In fact, the authors of [3] and [4] did not compare the computational efficiency of their methods with the method in [1] in their papers, because they take their methods being conceptually faster for granted. However, since the methods in [3], [4] are local-like methods, the computation time of each solution adjustment step is very short, but the improvement of the solution is limited. Hence their convergence rate will be degraded especially when the dimension of the ASABA problem is large. On the contrary, the computational complexity of our approach is less relevant to the size of the ASABA problem, because (i) the population size and number of iterations of the employed GA in stage 1 are fixed, (ii) the parameters s and l in stages 1 and 2, respectively, are fixed, and (iii) the structure of the ANN is also fixed. This is the reason why the computational efficiency of our algorithm can compete with the methods in [3], [4] in solving large-dimension ASABA problems. It is commonly understood that the comparisons based on CPU times may not be objective enough, however we can hardly obtain any analytical expression of the total consumed number of multiplications and additions of the methods in [1]–[4]. In fact, the CPU time is a commonly used tool for the comparisons of computational efficiency in similar subjects appearing in [5], [21], [22].

In previous comparisons, we have set the BER, $P_e = 10^{-4}$. It would be interesting to know how will the Quality-of-Service (QoS) requirement, i.e. various BER, affect the performance of our algorithm. Therefore, we have tested the five algorithms for K = 10, 20, 30, 40 and 50 with various P_e ranging from 10^{-2} to 10^{-6} using randomly generated 5000 sets of $\alpha_{k,n}$, $k = 1, \ldots, K$, $n = 1, \ldots, 256$, for each K. The conclusions on the performance for the five algorithms for various K are similar. A typical one is shown in Fig. 4, which corresponds to K = 40. The $\overline{abSNR}$ obtained by our algorithm is marked by "△" in Fig. 4. We see that the performance of our algorithm is the best among the five in all cases of P_e, and when the QoS level is required higher (i.e. the value of P_e is smaller), the performance of our algorithm is even better (i.e. smaller $\overline{abSNR}$ compared with the other four algorithms). This further demonstrates the superiority of the solution quality achieved by our algorithm.

5. Conclusion

We have proposed a computationally efficient three-stage OO approach to solve the large-dimension ASABA problem of multiuser OFDM system for a good enough solution. By looking into the insight of the ASABA problem (1), we reformulate it into (5) and develop an approximate objective function as well as a subtle representation scheme and a repair operator for the GA employed in Stage 1, which makes our OO approach possible in handling the huge discrete solution space as well as the constraints. The easily computed surrogate models employed in Stages 1 and 2 help resolve the computation complexity caused by the hard-to-evaluate objective function. These factors contribute most to the computational efficiency of our algorithm. Furthermore, we have demonstrated the superiority of our algorithm by comparing with four existing algorithms through numerous test cases in the aspects of solution quality and computational efficiency. More importantly, our approach has wide range of applications in resource allocation problems of wireless network and communication.

Acknowledgments

The authors would like to thank the anonymous reviewers for their constructive comments and valuable suggestions which are beneficial to improve the quality of the paper.

References

[1] C.Y. Wong, R.S. Cheng, K.B. Letaief, and R.D. Murch, "Multiuser OFDM with adaptive subcarrier, bit, and power allocation," IEEE J. Sel. Areas Commun., vol.17, pp.1747–1758, Oct. 1999.
[2] I. Kim, H.L. Lee, B. Kim, and Y.H. Lee, "Use of linear programming for dynamic subcarrier and bit allocation in multiuser OFDM," IEEE Trans. Veh. Technol., vol.55, no.4, pp.1195–1207, March 2006.
[3] M. Ergen, S. Coleri, and P. Varaiya, "QoS aware adaptive resource allocation techniques for fair scheduling in OFDMA based broadband wireless access systems," IEEE Trans. Broadcast., vol.49, no.4, pp.362–370, Dec. 2003.
[4] G. Zhang, "Subcarrier and bit allocation for real-time services in multiuser OFDM systems," Proc. IEEE International Conf. Communications, vol.5, pp.2985–2989, June 2004.
[5] D. Kivanc, G. Li, and H. Liu, "Computationally efficient bandwidth allocation and power control for OFDMA," IEEE Trans. Wirel. Commun., vol.2, no.6, pp.1150–1158, Oct. 2003.
[6] Z. Han, Z. Ji, and K.J.R. Liu, "Low-complexity OFDMA channel allocation with Nash bargaining solution fairness," Proc. IEEE Global Telecommunication Conf., vol.6, pp.3726–3731, Dec. 2004.
[7] S.M. Sait and H. Youssef, Iterative Computer Algorithms With Applications in Engineering: Solving Combinatorial Optimization Problems, IEEE Comput. Soc., Los Alamitos, CA, 1999.
[8] Y.C. Ho, O.C. Zhao, and Q.S. Jia, Ordinal optimization: Soft optimization for hard problems, Springer-Verlag, New York, 2007.
[9] Y.C. Ho, "An explanation of ordinal optimization: Soft computation of hard problems," Inf. Sci., vol.113, no.3-4, pp.169–192, 1999.
[10] T.W.E. Lau and Y.C. Ho, "Universal alignment probability and subset selection for ordinal optimization," J. Optim. Theory Appl., vol.39, no.3, pp.455–489, June 1997.
[11] S.K. Lai, R.S. Cheng, K. Ben Letaief, and R.D. Murch, "Adaptive trellis coded MQAM and power optimization for OFDM transmission," Proc. IEEE Vehicular Technology Conf. (VTC'99), Houston, TX, May 1999.
[12] E.K.P. Chong and S.H. Żak, An Introduction to Optimization, 2nd ed., Wiley, New York, 2001.
[13] C.T. Lin and C.S. George Lee, Neural Fuzzy System: A Neuro-Fuzzy Synergism to Intelligent Systems, Prentic-Hall, Englewood Cliffs, NJ, 1996.
[14] L.G. Alberto, Probability and Random Processes for Electrical Engineering, Second ed., Addison Wesley, 1994.
[15] M.T. Hagan and M. Menhaj, "Training feedforward networks with Marquardt algorithm," IEEE Trans. Neural Netw., vol.5, no.6, pp.989–993, Nov. 1994.
[16] G. Lera and M. Pinzolas, "Neighborhood based Levenberg—Marquardt algorithm for neural network training," IEEE Trans. Neural Netw., vol.13, no.5, pp.1200–1203, Sept. 2002.
[17] T.S. Rappaport, Wireless Communications: Principle and Practice, 2nd ed., Prentice Hall, 2002.
[18] S.Y. Lin, Y.C. Ho, and C.H. Lin, "An ordinal optimization theory based algorithm for solving the optimal power flow problem with discrete control variables," IEEE Trans. Power Syst., vol.19, no.1, pp.276–286, Feb. 2004.
[19] S.Y. Lin and S.C. Horng, "Application of an ordinal optimization algorithm to the wafer testing process," IEEE Trans. Syst. Man. Cybern. A, Syst. Humans, vol.36, no.6, pp.1229–1234, Nov. 2006.
[20] S.Y. Lin and S.C. Horng, "Ordinal optimization of G/G/1/K polling systems with k-limited service discipline," scheduled to appear in Dec. issue of J. Optim. Theory Appl.
[21] L.Y. Ou and Y.F. Chen, "An iterative multi-user bit and power allocation algorithm for DMT-based systems," IEICE Trans. Commun., vol.E88-B, no.11, pp.4259–4265, Nov. 2005.
[22] S.M. Lee and D.J. Park, "Fast optimal bit and power allocation based on the Lagrangian method for OFDM systems," IEICE Trans. Commun., vol.E89-B, no.4, pp.1346–1353, April 2006.

Shin-Yeu Lin was born in Taiwan, R.O.C. He received the B.S. degree in electronics engineering from National Chiao Tung University, the M.S. degree in electrical engineering from University of Texas at El Paso and the D.Sc. degree in systems science and mathematics from Washington University in St. Louis, Missouri, in 1975, 1979, and 1983, respectively. From 1984 to 1985, he was with Washington University working first as a Research Associate and then a Visiting Assistant Professor. From 1985 to 1986, he was with GTE Laboratory working as a Senior MTS. He joined the Department of Electrical and Control Engineering at National Chiao Tung University in 1987 and has been a Professor since 1992. His major research interests include wireless communication and network, ordinal optimization theory and applications, and distributed computations.

Jung-Shou Huang was born in Taiwan, R.O.C. He received the B.S. degree in electrical engineering from Tamkang University in 1996 and the M.S. degree in electrical and control engineering from National Chiao Tung University in 1998. He has been studying for his Ph.D. degree in electrical and control engineering from National Chiao Tung University since September 2002. Currently, he also works as a digital IC designer in Elan. His major research interests include optimization theory with applications, image processing and digital IC design.