

A FAST ANALYTICAL METHOD FOR OPTIMUM INTEGER BIT LOADING IN OFDM SYSTEMS*

M. HATAM¹ AND M. A. MASNADI-SHIRAZI^{2**}

Dept. of Communications and Electronics, School of Electrical and Computer Engineering, Shiraz University,
Shiraz, I. R. of Iran

Email: masnadi@shirazu.ac.ir

Abstract– In this paper the problem of minimizing the total transmitting power subject to a fixed total bit rate in OFDM systems is considered. Upper bounds on transmitting power and bit rate of each subcarrier can also be taken into account. In practice, the number of bits of each subcarrier should be integer and nonnegative. In this paper an analytical optimal solution is derived for the case of assuming the bits to be integer. Then, the solution is extended for the case of nonnegative integer bits and the cases in which we have constraints on the maximum power and bit rate of each subcarrier. In an OFDM system with N subcarriers the complexity of computing the proposed analytical solution is $O(N)$ which is lower than the computational complexity of existing algorithms. In addition to the mathematical proofs, computer simulations confirm that the proposed analytical solution is optimal and faster than the existing algorithms.

Keywords– Optimum bit loading, discrete multi-tone (DMT) modulation, discrete optimization, orthogonal frequency division multiplexing (OFDM)

1. INTRODUCTION

Orthogonal frequency division multiplexing (OFDM) is one of the most advantageous methods for digital data communications. In an OFDM system, data is transmitted through some orthogonal subcarriers. Compared to a single carrier communication system, an OFDM system requires lower transmission power due to lower bit rate at each subcarrier. Moreover, OFDM signaling is a very efficient technique for reducing inter-symbol interference (ISI) in channels with severe ISI. In a practical OFDM system, the attenuations and the noise powers at different subcarriers are not essentially equal and consequently to achieve the required bit error rate (BER) at each subcarrier, the bit rate and power of subcarriers may be different.

In this paper we consider the optimization problem of minimizing the total transmit power subject to a target total bit rate and bit error rate (BER) and upper bounds on transmitting power and bit rate of each subcarrier and propose an analytical method for solving this problem.

In practice, the number of bits of each subcarrier should be integer and nonnegative. Thus, integer optimization techniques should be used for solving the optimal bit loading problem. A common integer optimization algorithm for optimal bit loading is greedy algorithm. In a greedy algorithm, at each iteration one bit is allocated to the subcarrier that has the maximum decrease in required transmission power by receiving one extra bit. For example, in [1-7] greedy algorithms are proposed for the optimal integer bit loading in which the optimal solution is obtained by one-by-one bit loading. The computational complexity of these greedy algorithms are $O(N^2)$.

*Received by the editors July 10, 2014; Accepted April 13, 2015.

**Corresponding author

In [8], a modified greedy algorithm is proposed for optimal bit loading in which at the first step, the subcarriers are sorted based on their attenuations, and at the next steps the optimal bit loading is obtained by using greedy search in all possible bit switches. The complexity of this algorithm is $O(N \log_2(N))$.

In [9], a couple of group-by-group bit loading algorithms are proposed for the integer bit loading problem. Complexity study in [9] shows that the group-by-group bit loading algorithms are $O(N \log_2(N))$ and have lower computational complexities compared to the existing optimal algorithms which have the ability to set constraints on the bit rate and transmitting power of each subcarrier.

Many suboptimal algorithms are proposed for integer bit loading [10-14] aimed at reducing the computational complexity. However, as stated in [9], the complexity reduction in these algorithms is not significant and the order of computational complexity of these algorithms is not lower than that of the algorithm proposed in [9].

In this paper, an analytical solution is developed for the optimal bit loading in OFDM systems by using an analytical discrete optimization method and Lagrange multiplier analysis. Then, the analytical solution is extended for the cases of having constraints on the maximum power and bit rate of each subcarrier. The worst case computational complexity of the proposed analytical solution is $O(N)$ which is lower than the complexity of existing algorithms. The computer simulations confirm the mathematical proofs which show that the proposed method is faster than the existing algorithms.

In section 2, the formulation of bit loading problem is introduced and the related parameters are defined. In section 3, some mathematical tools that are used in this paper for solving the bit loading problem are introduced. In subsection 3-a, an analytical approach is introduced for finding minima and maxima of a discrete function including the global minimum/maximum. In subsection 3-b, Theorem 2 introduces a sufficient condition for optimality of the solution of dual Lagrange problem (strong duality). Then, according to Theorem 2 it is shown that the solution of main problem can be found by solving the unconstrained problem (8) and finding the optimal Lagrange multiplier (proof of strong duality based on Theorem 2 is provided in Appendix A-C). In subsections 3-c and 3-d, a brief introduction of *selection* and *bisection* algorithms is brought. These algorithms are used to reduce the computational complexity of finding the solution of bit loading problem. In section 4, first the closed form solution of unconstrained Lagrange problem is found (relation (17) for OIBL problem and (18) for ONIBL problem). Then, the solution of dual problem is found. According to strong duality which is proved in section 3 and Appendix A-c, this solution is optimal. The main differences between the proposed algorithm and the existing ones are explained in subsection 4-a. The results of computer simulations are explained in section 5.

The proofs of theorems, lemmas and the complexity of algorithms are provided in Appendix A.

2. OPTIMUM BIT LOADING

The aim of optimum bit loading (OBL) for an OFDM system with N subcarriers is to manage the bit rate of each subcarrier so that the total transmission power is minimized and sum of the loaded bits is equal to the OFDM symbol size B . To achieve the desired bit error rate (BER), using the gap-approximation, the number of bits of the i^{th} subcarrier is obtained as [15], [16]

$$b_i = \log_2 \left(\frac{|H_i|^2}{\Gamma \sigma_i^2} P_i + 1 \right) \quad (1)$$

where H_i is the complex gain of channel, σ_i^2 is the variance of the additive noise of channel, and Γ is a constant obtained from gap analysis in order to achieve the desired BER. The value of Γ is obtained in terms of BER as [17]:

$$\Gamma = -\ln(5BER)/1.5$$

According to (1) the power of i^{th} subcarrier can be written in terms of the number of bits of subcarrier as

$$P_i(b_i) = \frac{\Gamma \sigma_i^2}{|H_i|^2} (2^{b_i} - 1) \quad (2)$$

Let $\mathbf{b} = [b_1, b_2, \dots, b_N]^T$ be the bit loading vector. The problem can be formulated as

$$\begin{aligned} \text{Minimize } \sum_{i=1}^N P_i(b_i) \quad \text{Subject to: } \sum_{i=1}^N b_i = B \\ b_i \in \mathbf{D}_i, \quad i = 1, \dots, N \end{aligned} \quad (3)$$

It should be noted that in (3), the BER constraint is also considered because the value of Γ depends on the desired BER of each subcarrier and the effect of BER is included in the cost function of problem. In practice, since the number of bits is integer and nonnegative we have $\mathbf{D}_i = \mathbb{Z}^+$. In this paper first we derive the solution of problem for $\mathbf{D}_i = \mathbb{Z}$ and name the corresponding problem as optimum integer bit loading (OIBL). Then we modify the solution for $\mathbf{D}_i = \mathbb{Z}^+$ and name the problem optimum nonnegative integer bit loading (ONIBL).

Sometimes, for power spectral density compatibility, the number of bits of each subcarrier is constrained to upper bounds as [18]

$$b_i \leq u_i, \quad i = 1, 2, \dots, N \quad (4)$$

where the upper bound u_i is obtained from

$$u_i = \min \left(b_{\max}, \left\lfloor \log_2 \left(\frac{|H_i|^2}{\Gamma \sigma_i^2} \bar{P} + 1 \right) \right\rfloor \right) \quad (5)$$

where \bar{P} is the maximum allowable transmission power of each subcarrier and b_{\max} is the maximum allowable number of bits of each subcarrier and $\lfloor \cdot \rfloor$ denotes the floor function. Clearly, according to the constraint $\sum_{i=1}^N b_i = B$, we must have $\sum_{i=1}^N u_i \geq B$.

3. DISCRETE OPTIMIZATION TOOLS

a) Analytical discrete optimization approach

In [19] we have proposed an analytical method for finding the local minima and maxima of a discrete function. Here, we repeat some parts of this method.

Definition of Local Minimum / Maximum for a Discrete Function:

Let $f[n]$ be a discrete function defined on $\mathbf{D} \subset \mathbb{Z}$. We define n^* as a local minimum of $f[n]$ if n^* , $n^* - 1$ and $n^* + 1$ are in \mathbf{D} and we have $f[n^*] \leq f[n^* - 1]$ and $f[n^*] \leq f[n^* + 1]$. Similarly define n^* as a local maximum of $f[n]$ if n^* , $n^* - 1$ and $n^* + 1$ are in \mathbf{D} and we have $f[n^*] \geq f[n^* - 1]$ and $f[n^*] \geq f[n^* + 1]$.

Theorem 1: Suppose that $f[n]$ is a discrete function defined on the integer interval $I \subset \mathbb{Z}$ and $f_r(x)$ is a continuous real function defined on the connected interval $I_r \subset \mathbb{R}$ so that $I \subset I_r$ and at each integer point in I_r such as n' we have $f_r(n') = f[n']$. Let x_1, x_2, \dots, x_m be all the solutions of the equation $f_r(x) - f_r(x - 1) = 0$. The set \mathbf{A} consisting of the floors of all x_i s ($i = 1, 2, \dots, m$) plus integer $(x_i - 1)$ s (if any) contains all the local minima and maxima of $f[n]$.

Remark 1: The global minimum (maximum) of a discrete function defined on $[n_1, n_2]$ is a local minimum (maximum) or a boundary point (n_1 or n_2).

As a simple proof, if the global optimum is a boundary point, Remark 1 is satisfied. Otherwise, according to the definition of local optimums, the global optimum is a local optimum.

Remark 2 : When $f_r(x)$ is a unimodal function, $f[n]$ is unimodal (has only one minimum or maximum) or has two successive minima (or two successive maxima) as n_1^* and $n_1^* - 1$ for which $f[n_1^*] = f[n_1^* - 1]$.

b) Lagrange multiplier analysis

Theorem 2: Consider the constrained optimization problem

$$\text{Minimize}_{\mathbf{x} \in \mathbf{D}} \{G(\mathbf{x})\} \quad \text{Subject to: } C(\mathbf{x}) = c \quad (6)$$

where \mathbf{x} is a N -dimensional vector and \mathbf{D} is a set of N -dimensional numbers (convex or nonconvex). Assume that $\mathbf{x}^*(\lambda)$ is the solution of the unconstrained problem:

$$\text{Minimize}_{\mathbf{x} \in \mathbf{D}} \{G(\mathbf{x}) + \lambda C(\mathbf{x})\} \quad (7)$$

If a $\lambda^* \in \mathbb{R}$ can be found that satisfies $C(\mathbf{x}^*(\lambda^*)) = c$ then $\mathbf{x}^*(\lambda^*)$ is the solution of (6).

The proof of similar theorem in the case of inequality constraint ($C(\mathbf{x}) \leq c$) is proposed in [20] but for the case of equality constraint (Theorem 2) we have provided the proof in Appendix A.

According to Theorem 2, to solve the problem (3) at first we should find

$$\text{Minimize}_{\mathbf{b}} \left\{ \sum_{i=1}^N P_i(b_i) + \lambda \sum_{i=1}^N b_i \right\}, \quad (8)$$

$$b_i \in \mathbf{D}_i, \quad i = 1, \dots, N$$

According to Remark 2, each b_i in (8) may have two optimal values for each λ . We denote the optimal values of b_i at each λ by the set function $\{b_i^*(\lambda)\}$.

The value of λ^* (the optimum λ) can be found by solving

$$\sum_{i=1}^N \{b_i^*(\lambda)\} = B \quad (9)$$

Then, the optimal solution of (3) can be obtained by substituting the optimal λ in $\{b_i^*(\lambda)\}$ s. It can be shown that for OIBL and ONIBL problems, the condition of Theorem 2 is satisfied and the resultant solution from (8) and (9) is the same as that of (3) (see the proof of strong duality in Appendix A).

c) Selection algorithm

Let \mathbf{T} be a set of N numbers. Define a rank for each member of \mathbf{T} so that the largest member has the rank 1, the smallest member has the rank N and in the sorted list of members of \mathbf{T} , the i^{th} number of list has the rank i . However, to find the number with rank i in the set \mathbf{T} it is not necessary to sort the list.

Selection algorithm finds the number with rank i in a set of N numbers. Best selection algorithms are $O(N)$ while best sorting algorithms are $O(N \log N)$. Using a similar $O(N)$ algorithm we can also find the subset of i largest (or smallest) numbers of a set of N numbers [21-23].

Some definitions:

Define $\psi(i, \mathbf{T})$ as the i^{th} large member (the number with rank i) in the set \mathbf{T} .

Define $\Phi(i, \mathbf{T})$ as the subset of i largest members of the set \mathbf{T} and define $\Phi'(i, \mathbf{T})$ as the subset of i smallest members of the set \mathbf{T} .

Define $\bar{\Phi}(i, j, \mathbf{T})$ as the subset of members with ranks between i and j in the set \mathbf{T} . We have

$$\bar{\Phi}(i, j, \mathbf{T}) = \Phi(j, \mathbf{T}) - \Phi(i - 1, \mathbf{T}), \quad j > i$$

Define $\mathfrak{R}(x, \mathbf{T})$ as the rank of x in the set \mathbf{T} .

d) Discrete bisection approach

Bisection algorithm is a well-known algorithm for finding the roots of real functions. Here we present a discrete version of this algorithm. Let $f[n]$ be a non-increasing discrete function and assume that we have $f[n_1] \geq 0$, $f[n_2] \leq 0$ and $n_2 - n_1 = N > 0$. In the discrete version of algorithm our goal is to find n^* such that

$$f[n^*] \geq 0 \text{ and } f[n^* + 1] \leq 0 \tag{10}$$

Algorithm 1: Discrete Bisection Algorithm

Step 1: Set $\bar{n} = \lfloor (n_1 + n_2) / 2 \rfloor$.

Step 2: If (10) is satisfied for $n^* = \bar{n}$ then stop the algorithm and set $n^* = \bar{n}$. Otherwise:

If $f[\bar{n}] > 0$ set $n_1 = \bar{n} + 1$ and go to Step 1.

If $f[\bar{n}] < 0$ set $n_2 = \bar{n} - 1$ and go to Step 1.

Stop

In worst case, this algorithm requires $\lfloor \log_2(N) \rfloor + 1$ comparison operations (see Appendix A). For non-decreasing discrete functions, directions of inequalities in Algorithm 1 should be reversed.

4. SOLVING THE PROBLEM

In (8) only the i^{th} term of each summation depends on b_i and the other terms are independent of b_i . Thus, $\{b_i^*(\lambda)\}$ can be obtained from

$$\{b_i^*(\lambda)\} = \arg \min_{b_i \in \mathbb{Z}} \{G(b_i, \lambda)\} \tag{11}$$

where

$$G(b_i, \lambda) = P_i(b_i) + \lambda b_i, \quad i = 1, \dots, k \tag{12a}$$

$$= \frac{\Gamma \sigma_i^2}{|H_i|^2} (2^{b_i} - 1) + \lambda b_i, \quad i = 1, \dots, k \tag{12b}$$

According to (12), $G(b_i, \lambda)$ is a unimodal function of b_i for $\lambda < 0$ and it is a strictly increasing function of b_i for $\lambda \geq 0$. Thus, for $\lambda \geq 0$ we have $\{b_i^*(\lambda)\} = -\infty, \forall i$ and consequently $\sum_{i=1}^N \{b_i^*(\lambda)\} = -\infty \neq B$. It means that we must have $\lambda^* < 0$. Therefore, we consider $\{b_i^*(\lambda)\}$ only for $\lambda < 0$.

Define the real function

$$G(\bar{b}_i, \lambda) = P_i(\bar{b}_i) + \lambda \bar{b}_i \tag{13}$$

where \bar{b}_i is a real variable and $P_i(\bar{b}_i)$ is a real function that at integer \bar{b}_i s is equal to $P_i(b_i)$ (as declared in Theorem 1). To find $\{b_i^*(\lambda)\}$ according to Theorem 1 we have

$$\begin{aligned} G(\bar{b}_i, \lambda) - G(\bar{b}_i - 1, \lambda) &= 0 \\ P_i(\bar{b}_i) + \lambda \bar{b}_i - (P_i(\bar{b}_i - 1) + \lambda(\bar{b}_i - 1)) &= 0 \\ P_i(\bar{b}_i) - P_i(\bar{b}_i - 1) &= -\lambda \end{aligned} \tag{14}$$

For convenience we define

$$C_i = \frac{\Gamma \sigma_i^2}{|H_i|^2} \quad (15)$$

Substituting (2) and (15) in (14) yields

$$\begin{aligned} C_i \left(2^{\bar{b}_i} - 2^{\bar{b}_i - 1} \right) &= -\lambda \\ C_i 2^{\bar{b}_i - 1} &= -\lambda \\ \bar{b}_i(\lambda) &= \log_2(-2\lambda / C_i), \lambda < 0 \end{aligned} \quad (16)$$

According to Theorem 1 and (16), $\{b_i^*(\lambda)\}$ which is the minimum of the discrete function $G(b_i, \lambda)$, for $\lambda < 0$ is obtained from

$$\{b_i^*(\lambda)\} = \begin{cases} \log_2(-2\lambda / C_i) \text{ or } \log_2(-2\lambda / C_i) - 1 & \text{if } \log_2(-2\lambda / C_i) \text{ is integer} \\ \lfloor \log_2(-2\lambda / C_i) \rfloor & \text{otherwise} \end{cases} \quad (17)$$

In an ONIBL problem in which we have the constraints

$$0 \leq b_i \leq u_i, \quad i = 1, 2, \dots, N$$

to achieve the target rate we must have $\sum_{i=0}^N u_i \geq B$.

Since $G(b_i, \lambda)$ is a unimodal function of b_i for $\lambda < 0$, if the minimum of $G(b_i, \lambda)$ is outside the interval $0 \leq b_i \leq u_i$ then according to Remark 1 its minimum in $[0, u_i]$ is 0 or u_i . A unimodal function is decreasing at the left-hand side of its minimum and increasing at the right-hand side of its minimum. Thus, in the ONIBL problem we have

$$\{b_i^*(\lambda)\} = \begin{cases} 0 & \text{if } \lfloor \log_2(-2\lambda / C_i) \rfloor \leq 0 \\ u_i & \text{if } \lfloor \log_2(-2\lambda / C_i) \rfloor \geq u_i \\ \log_2(-2\lambda / C_i) \text{ or } \log_2(-2\lambda / C_i) - 1 & \text{if } \log_2(-2\lambda / C_i) \text{ is an integer in } [1, u_i - 1] \\ \lfloor \log_2(-2\lambda / C_i) \rfloor & \text{otherwise} \end{cases} \quad (18)$$

In this case modify the definition of $\bar{b}_i(\lambda)$ as

$$\bar{b}_i(\lambda) = \begin{cases} 0 & \text{if } \lfloor \log_2(-2\lambda / C_i) \rfloor \leq 0 \\ u_i & \text{if } \lfloor \log_2(-2\lambda / C_i) \rfloor \geq u_i \\ \log_2(-2\lambda / C_i) & \text{otherwise} \end{cases} \quad (19)$$

It should be noted that since $G(b_i, \lambda)$ is a unimodal function, according to Theorem 1 and Remark 2, (17) and (18) are global minimum of $G(b_i, \lambda)$ in OIBL and ONIBL problems, respectively. Define

$$\bar{B}(\lambda) = \sum_{i=1}^N \bar{b}_i(\lambda) \quad (20)$$

$$\{B^*(\lambda)\} = \sum_{i=1}^N \{b_i^*(\lambda)\} \tag{21}$$

Typical plots of $\{b_i^*(\lambda)\}$, $\{B^*(\lambda)\}$, $\bar{b}_i(\lambda)$ and $\bar{B}(\lambda)$ are shown in Fig. 1 and Fig. 2. The optimum λ (λ^* defined in subsection 3-b) can be found by solving

$$\{B^*(\lambda)\} = B \tag{22}$$

Substituting λ in $\{b_i^*(\lambda)\}$ s, b_i^* s are obtained.

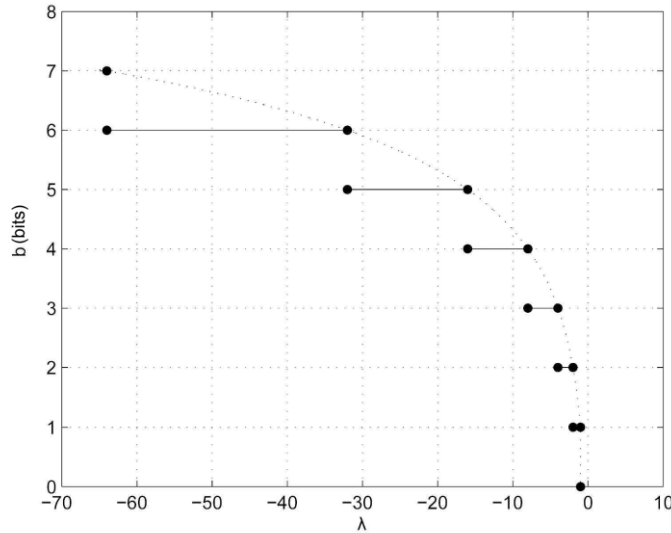


Fig. 1. Typical diagram of $b_i^*(\lambda)$ (solid) and $\bar{b}_i(\lambda)$ (dotted) versus λ

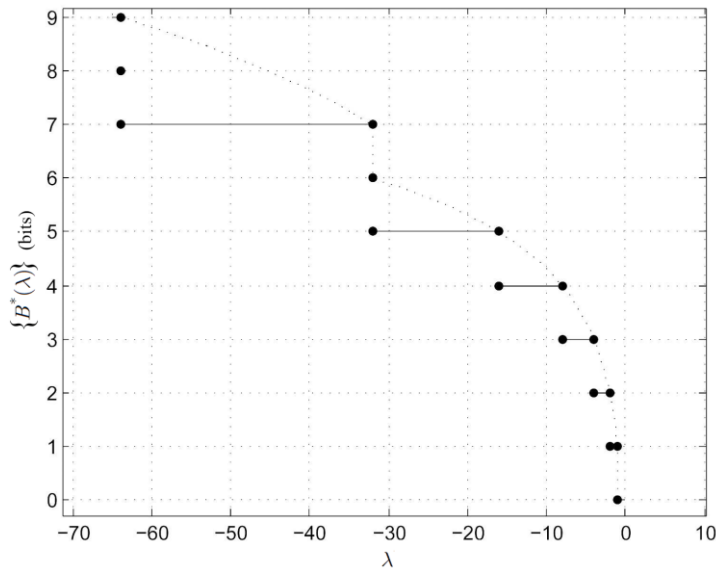


Fig. 2. Typical diagram of $B^*(\lambda)$ (solid) and $\bar{B}(\lambda)$ (dotted) versus λ

Some definitions:

Define λ^c as a critical λ of $\{b_i^*(\lambda)\}$ if $\{b_i^*(\lambda)\}$ has a jump and two different values at λ^c .

Define λ^c as a critical λ of $\{B^*(\lambda)\}$ if $\{B^*(\lambda)\}$ has a jump and two or more than two values at λ^c .

Since $\{B^*(\lambda)\} = \sum_{i=1}^N \{b_i^*(\lambda)\}$, each critical λ of $\{b_i^*(\lambda)\}$ s is a critical λ of $\{B^*(\lambda)\}$.

Define

$$b_i^*(\lambda)_U = \lfloor \log_2(-2\lambda/C_i) \rfloor \tag{23a}$$

$$B^*(\lambda)_U = \sum_{i=1}^N b_i^*(\lambda)_U \tag{23b}$$

According to (17) and (18) when $\log_2(-2\lambda/C_i)$ is non-integer we have $b_i^*(\lambda)_U = \{b_i^*(\lambda)\}$ and when it is integer (at critical λ s) $b_i^*(\lambda)_U$ is equal to the upper value of $\{b_i^*(\lambda)\}$. Similarly, $B^*(\lambda)_U$ at critical λ s is equal to the largest value of $\{B^*(\lambda)\}$.

For each $i \in \{1, 2, \dots, N\}$ and any integer j , define $\lambda_{i,j}^c$ as a critical λ at which $\bar{b}_i(\lambda) = j$:

$$\bar{b}_i(\lambda) = \log_2(-2\lambda/C_i) = j$$

$$\lambda_{i,j}^c = -C_i 2^{j-1} \tag{24}$$

In the case of having the constraints $0 \leq b_i \leq u_i$ and $\bar{b}_i(\lambda)$ defined as (19), $\lambda_{i,j}^c$ is only valid for $0 \leq j \leq u_i$ and outside this interval $\{b_i^*(\lambda)\}$ does not have any critical λ .

We name any interval of λ at which $\{B^*(\lambda)\}$ is constant, as a step. A step of $\{B^*(\lambda)\}$ may be a singular point as illustrated in Figs. 3 and 4 (see also Lemma 2).

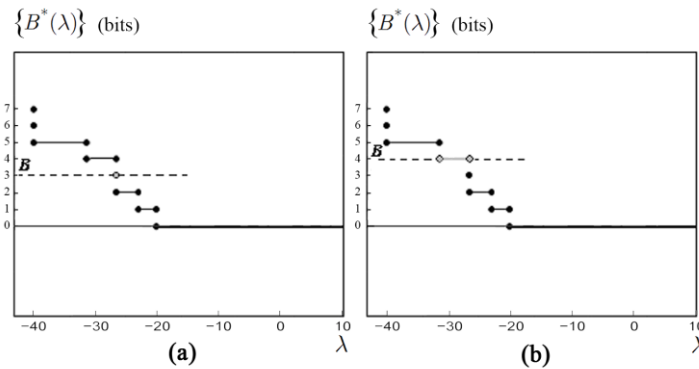


Fig. 3. Intersection of $B^*(\lambda)$ with the horizontal line $f(\lambda) = B$ (a) In multi solution case the intersection is a singular point (b) In unique solution case the intersection is a nonsingular step

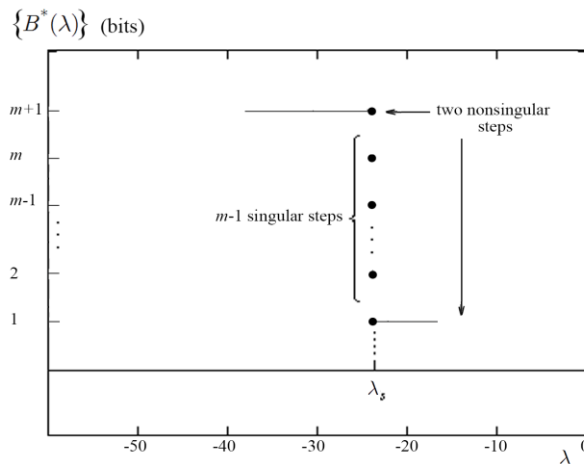


Fig. 4. Typical diagram of $B^*(\lambda)$ about the critical point λ_s when m number of $b_i^*(\lambda)$ s ($m \geq 2$) have two values at λ_s

Lemma 1: For each $i, k \in \{1, 2, \dots, N\}$ and any integer j , $b_k^*(\lambda)$ at most has one critical λ in the interval $[\lambda_{i,j+1}^c, \lambda_{i,j}^c]$.

Lemma 2: The OIBL and ONIBL problems have the following properties:

If the problem has more than one optimal solution, then the optimal λ is a unique critical λ as λ_s and all the solutions correspond to a singular point in the diagram of $\{B^*(\lambda)\}$ and the number of solutions of the problem is $\binom{m}{l}$ where m is the number of $\{b_i^*(\lambda)\}$ s that has two values at λ_s ($m \geq 2$) and $l = B^*(\lambda_s)_U - B$. If the problem has a unique solution then the solution occurs at a nonsingular step of $\{B^*(\lambda)\}$ which contains infinite values of optimum λ (see Fig. 3).

Remark 3: It is obvious from (16)-(21) that $\{b_i^*(\lambda)\}$, $\{B^*(\lambda)\}$, $\bar{b}_i(\lambda)$ and $\bar{B}(\lambda)$ are non-increasing with λ .

Here, at first we solve the OIBL problem and then modify the solution for ONIBL. Let λ^* be an optimum λ and b_i^* be an optimum value of b_i related to λ^* in a OIBL problem. We would have $\lambda_{i,b_i^*+1}^c \leq \lambda^* \leq \lambda_{i,b_i^*}^c$ because outside this interval $\{b_i^*(\lambda)\} \neq b_i^*$. First we assume that the solution of the OIBL problem occurs at a nonsingular step of $\{B^*(\lambda)\}$ and all the b_i s have one optimal value (see Lemma 2). In this case since $\{B^*(\lambda^*)\} = B$, $\lambda_{i,b_i^*+1}^c \leq \lambda^* \leq \lambda_{i,b_i^*}^c$ and $\{B^*(\lambda)\}$ is non-increasing with λ , we have $B^*(\lambda_{i,b_i^*}^c)_U \leq B$. Also, $\{B^*(\lambda)\}$ has a jump at $\lambda_{i,b_i^*+1}^c$ and we have $B^*(\lambda_{i,b_i^*+1}^c)_U > B$. Therefore, b_i^* is the largest integer number like j that satisfies $B^*(\lambda_{i,j}^c)_U \leq B$. i.e.

$$b_i^* = \max \left\{ j : B^*(\lambda_{i,j}^c)_U = \sum_{n=1}^N b_n^*(\lambda_{i,j}^c)_U \leq B \text{ and } j \in \mathbb{Z} \right\} \quad (25)$$

Substituting (22) and (23) in (25) yields

$$\begin{aligned} b_i^* &= \max \left\{ j : \sum_{n=1}^N \lfloor \log_2(C_i / C_n) + j \rfloor \leq B \text{ and } j \in \mathbb{Z} \right\} \\ &= \max \left\{ j : j \leq \frac{B}{N} - \frac{1}{N} \sum_{n=1}^N \lfloor \log_2(C_i / C_n) \rfloor \text{ and } j \in \mathbb{Z} \right\} \end{aligned} \quad (26)$$

According to definition of floor, from (26) we conclude

$$b_i^* = \left\lfloor \frac{B}{N} - \frac{1}{N} \sum_{n=1}^N \lfloor \log_2(C_i / C_n) \rfloor \right\rfloor \quad (27)$$

In the case that the optimal λ is critical, the problem has more than one solution and according to Lemma 2, in this case at least two numbers of b_i s have two optimal values and in order to satisfy $\sum_{i=1}^N b_i = B$, for l number of b_i s we should select the lower value and for the remaining b_i s the upper values should be selected as declared in Lemma 2. Furthermore, (27) can be expressed in another form with lower computational complexity (see Theorem 3).

Theorem 3: The optimal solution of OIBL problem is obtained from the following relation:

$$b_i^* = \begin{cases} \tilde{b} - \lfloor \log_2(C_i) \rfloor + 1 & \text{if } \rho_i \in \tilde{\mathbf{P}} \\ \tilde{b} - \lfloor \log_2(C_i) \rfloor & \text{otherwise} \end{cases} \quad (28)$$

where

$$\tilde{b} = \left\lfloor \frac{B + B_0}{N} \right\rfloor, \rho_i = \log_2(C_i) - \lfloor \log_2(C_i) \rfloor,$$

$$B_0 = \sum_{i=1}^N \lfloor \log_2(C_i) \rfloor, \tilde{B} = B + B_0 - N\tilde{b}$$

and

$$\tilde{\mathbf{P}} = \Phi'(\tilde{B}, \{\rho_i\}_{i=1}^N)$$

In the case that some of b_i s have two optimal values, the corresponding ρ_i s are equal and $\tilde{\mathbf{P}}$ which is the set of \tilde{B} smallest members of $\{\rho_n\}_{n=1}^N$ contains multiple combinations and all the possible combinations of $\tilde{\mathbf{P}}$ are acceptable in (28). This results in $\binom{m}{l}$ possible solutions for the problem where m is the number of b_i s that have two optimal values (have equal ρ_i s) and $l = (N\tilde{b} + N - B_0) - B$. ||

The set $\tilde{\mathbf{P}}$ can be found by using the selection algorithm whose complexity is $O(N)$. Since $\tilde{\mathbf{P}}$ and \tilde{b} are independent of i , the complexity of computing all b_i^* s from (28) is altogether $O(N)$.

Now, we modify the analytical solution (28) for the ONIBL problem.

Some Definitions:

Define $\mathbf{S} = \{1, 2, \dots, N\}$ as the set of all subcarriers.

For any set \mathbf{T} define $|\mathbf{T}|$ as the number of members of \mathbf{T} .

Define \mathbf{S}^+ as the set of subcarriers for which $b_i^* > 0$ and define $N^+ = |\mathbf{S}^+|$.

In the case that the number of bits is constrained to the upper bound as (4), define \mathbf{S}^u as the set of subcarriers for which $b_i^* = u_i$ and define $N^u = |\mathbf{S}^u|$.

Define $\mathbf{S}' = \mathbf{S}^+ - \mathbf{S}^u$ and $N' = N^+ - N^u$.

Lemma 3: The solution of ONIBL problem for the subcarriers in \mathbf{S}' is the solution of OIBL problem for the subcarriers in \mathbf{S}' and symbol size $B - \sum_{i \in \mathbf{S}^u} u_i$, i.e. the solution of the following problem

$$\min_{\mathbf{b}} \sum_{i \in \mathbf{S}'} P_i(b_i) \quad \text{Subject to: } \sum_{i \in \mathbf{S}'} b_i = B - \sum_{i \in \mathbf{S}^u} u_i, \quad b_i \in \mathbb{Z} \quad (29)$$

Lemma 4: Assume that in an ONIBL problem we have the constraints $0 \leq b_i \leq u_i$ and let λ^* be an optimum λ for the corresponding Lagrange unconstrained problem (problem (8)) for which the constraints $0 \leq b_i \leq u_i$ are applied to \mathbf{D}_i . So for this problem:

- The optimal value of b_i is $b_i^* = 0$ if and only if $\lambda^* > \lambda_{i,1}^c$.
- The optimal value of b_i is $b_i^* = u_i$ if and only if $\lambda^* < \lambda_{i,u_i}^c$. ||

According to Lemma 4 we have

$$\mathbf{S}^+ = \{i : \lambda^* \leq \lambda_{i,1}^c\} \quad (30)$$

$$\mathbf{S}^u = \{i : \lambda^* < \lambda_{i,u_i}^c\} \quad (31)$$

Define

$$\mathbf{\Lambda}^1 = \{\lambda_{i,1}^c\}_{i=1}^N \quad (32)$$

$$\mathbf{\Lambda}^u = \{\lambda_{i,u_i}^c\}_{i=1}^N \quad (33)$$

$$\mathbf{\Lambda} = \mathbf{\Lambda}^1 \cup \mathbf{\Lambda}^u \quad (34)$$

$$\mathbf{S}^+(\lambda) = \{i : \lambda_{i,1}^c \geq \lambda\} \quad (35)$$

$$\mathbf{S}^u(\lambda) = \{i : \lambda_{i,u_i}^c > \lambda\} \quad (36)$$

$$\mathbf{S}'(\lambda) = \mathbf{S}^+(\lambda) - \mathbf{S}^u(\lambda) \quad (37)$$

$$N'(\lambda) = |\mathbf{S}'(\lambda)| \quad (38)$$

$$B'(\lambda) = \sum_{i \in \mathbf{S}'(\lambda)} \log_2(C_i) \quad (39)$$

$$B^u(\lambda) = \sum_{i \in \mathbf{S}^u(\lambda)} u_i \quad (40)$$

To find an optimum λ , we can solve (22) using the bisection algorithm, but to reduce the complexity, at first we solve $\bar{B}(\lambda) = B$.

Define $\bar{\lambda}^*$ as the λ at which $\bar{B}(\lambda) = B$. In the case that $\bar{B}(\lambda)$ has a jump such that $\bar{B}(\lambda) = B$ has no solution, define $\bar{\lambda}^*$ as the critical lambda λ_c at which $\lim_{\lambda \rightarrow \lambda_c^-} \bar{B}(\lambda) < B$ and $\lim_{\lambda \rightarrow \lambda_c^+} \bar{B}(\lambda) > B$.

Define $\bar{B}' = B'(\bar{\lambda}^*)$, $\bar{B}^u = B^u(\bar{\lambda}^*)$, $\bar{\mathbf{S}}' = \mathbf{S}'(\bar{\lambda}^*)$, $\bar{\mathbf{S}}^+ = \mathbf{S}^+(\bar{\lambda}^*)$, $\bar{\mathbf{S}}^u = \mathbf{S}^u(\bar{\lambda}^*)$ and $\bar{N}' = |\bar{\mathbf{S}}'|$.

Lemma 5: In the ONIBL problem we have

a)

$$\bar{B}(\lambda) - N'(\lambda) \leq \{B^*(\lambda)\} \leq \bar{B}(\lambda), \forall \lambda \quad (41)$$

b)

$$\bar{B}(\lambda) = N'(\lambda) \log_2(-2\lambda) - B'(\lambda) + B^u(\lambda), \forall \lambda \quad (42)$$

c) If $\bar{\lambda}^*$ is not critical then

$$\bar{\lambda}^* = -2^{(B + \bar{B}' - \bar{B}^u) / \bar{N}' - 1} \quad (43)$$

d)

$$b_i^* - 1 \leq \{b_i^*(\bar{\lambda}^*)\} \leq b_i^*, \forall i \quad (44)$$

To solve $\bar{B}(\lambda) = B$, at first we can find $\bar{\mathbf{S}}^+$ and $\bar{\mathbf{S}}^u$ and then find $\bar{\lambda}^*$ using (38)-(40) and (43). To find $\bar{\mathbf{S}}^+$ and $\bar{\mathbf{S}}^u$ according to (30)-(36) and definition of $\bar{\mathbf{S}}^+$ and $\bar{\mathbf{S}}^u$, we can apply Algorithm 1 to the discrete function $f[n] = \bar{B}(\psi(n, \Lambda)) - B$ and substitute $\lambda = \psi(n^*, \Lambda)$ in (35) and (36). Since at each iteration of Algorithm 1 n is in the interval $[n_1, n_2]$, in order to reduce the complexity of computing $\psi(n, \Lambda)$ we can limit the set Λ as $\bar{\Phi}(n_1, n_2, \Lambda)$, i.e.

$$\psi(n, \Lambda) = \psi(n - n_1 + 1, \bar{\Phi}(n_1, n_2, \Lambda)) \quad (45)$$

To decrease the complexity of computing $B'(\psi(n, \Lambda))$ and $B^u(\psi(n, \Lambda))$, at each iteration of the Algorithm 1 for solving $\bar{B}(\psi(n, \Lambda)) - B = 0$ we can compute only the difference of the summations from the previous iteration (denoted by $\Delta \bar{B}'$ and $\Delta \bar{B}^u$). Based on the above explanations we come to the following low complexity algorithm.

Algorithm 2: Solving $\bar{B}(\lambda) = B$ and finding $\bar{\mathbf{S}}^+$, $\bar{\mathbf{S}}^u$ and $\bar{\lambda}^*$

Step 1- Set $N_1 = 1$, $N_2 = |\Lambda|$ and $\bar{\Lambda} = \Lambda$ (defined in (34)), $\bar{\mathbf{S}}^+ = \mathbf{S}$, $\bar{\mathbf{S}}^u = \mathbf{S}$, $\bar{\mathbf{S}}' = \{\}$, $\bar{N}' = 0$

$$B' = 0, B^u = \sum_{i=1}^N u_i \text{ and } \bar{\lambda} = \min(\bar{\Lambda}).$$

Step 2- Set $\bar{N} = \lfloor (N_1 + N_2) / 2 \rfloor$, $\bar{\lambda}_{old} = \bar{\lambda}$, $\bar{\lambda} = \psi(\bar{N} - N_1 + 1, \bar{\Lambda})$, $\bar{\lambda}^- = \psi(\bar{N} - N_1 + 2, \bar{\Lambda})$.

If $\bar{\lambda} < \bar{\lambda}_{old}$ compute $\Delta\bar{\mathbf{S}}^+ = \{i : \lambda_{i,1}^c \in \bar{\Lambda} \text{ and } \bar{\lambda} \leq \lambda_{i,1}^c\}$, $\Delta\bar{\mathbf{S}}^u = \{i : \lambda_{i,u_i}^c \in \bar{\Lambda} \text{ and } \bar{\lambda} < \lambda_{i,u_i}^c\}$
and then set $\Delta\bar{\mathbf{S}}'_+ = \Delta\bar{\mathbf{S}}^+ - \Delta\bar{\mathbf{S}}^u$, $\Delta\bar{\mathbf{S}}'_- = \Delta\bar{\mathbf{S}}^u - \Delta\bar{\mathbf{S}}^+$, $\Delta\bar{B}' = \sum_{i \in \Delta\bar{\mathbf{S}}'_+} \log_2(C_i) - \sum_{i \in \Delta\bar{\mathbf{S}}'_-} \log_2(C_i)$,

$$\Delta\bar{B}^u = \sum_{i \in \Delta\bar{\mathbf{S}}^u} \log_2(C_i), \bar{B}^u = \bar{B}^u + \Delta\bar{B}^u, \bar{B}' = \bar{B}' + \Delta\bar{B}', \bar{N}' = \bar{N}' + |\Delta\bar{\mathbf{S}}'_+| - |\Delta\bar{\mathbf{S}}'_-|.$$

If $\bar{\lambda} > \bar{\lambda}_{old}$ compute $\Delta\bar{\mathbf{S}}^+ = \{i : \lambda_{i,1}^c \in \bar{\Lambda} \text{ and } \bar{\lambda} \geq \lambda_{i,1}^c\}$, $\Delta\bar{\mathbf{S}}^u = \{i : \lambda_{i,u_i}^c \in \bar{\Lambda} \text{ and } \bar{\lambda} \geq \lambda_{i,u_i}^c\}$
and then set $\Delta\bar{\mathbf{S}}'_+ = \Delta\bar{\mathbf{S}}^u - \Delta\bar{\mathbf{S}}^+$, $\Delta\bar{\mathbf{S}}'_- = \Delta\bar{\mathbf{S}}^+ - \Delta\bar{\mathbf{S}}^u$, $\Delta\bar{B}' = \sum_{i \in \Delta\bar{\mathbf{S}}'_+} \log_2(C_i) - \sum_{i \in \Delta\bar{\mathbf{S}}'_-} \log_2(C_i)$,

$$\Delta\bar{B}^u = \sum_{i \in \Delta\bar{\mathbf{S}}^u} \log_2(C_i), \bar{B}^u = \bar{B}^u - \Delta\bar{B}^u, \bar{B}' = \bar{B}' + \Delta\bar{B}', \bar{N}' = \bar{N}' + |\Delta\bar{\mathbf{S}}'_+| - |\Delta\bar{\mathbf{S}}'_-|.$$

Similarly, in the above relations replace $\bar{\lambda}$ by $\bar{\lambda}^-$ to find $B'(\bar{\lambda}^-)$, $B^u(\bar{\lambda}^-)$, and $N'(\bar{\lambda}^-)$ and save them as \bar{B}'^- , \bar{B}^{u-} , and \bar{N}'^- , respectively.

Compute $\bar{B}(\bar{\lambda}) = \bar{N}' \log_2(-2\bar{\lambda}) - \bar{B}' + \bar{B}^u$ and $\bar{B}(\bar{\lambda}^-) = \bar{N}'^- \log_2(-2\bar{\lambda}^-) - \bar{B}'^- + \bar{B}^{u-}$.

Step 3- If $\bar{B}(\bar{\lambda}) \leq B$ and $\bar{B}(\bar{\lambda}^-) > B$ go to step 4; otherwise:

If $\bar{B}(\bar{\lambda}) > B$ set $N_2 = \bar{N} - 1$, $\bar{\Lambda} = \Phi(\bar{N} - N_1, \bar{\Lambda})$ and go to step 2.

If $\bar{B}(\bar{\lambda}^-) < B$ set $N_1 = \bar{N} + 1$, $\bar{\Lambda} = \Phi(N_2 - \bar{N}, \bar{\Lambda})$ and go to step 2.

Step 4- Compute $\bar{\lambda}^*$ from (43). If $\bar{B}(\bar{\lambda}^*) \neq B$ (which means that the actual $\bar{\lambda}^*$ is critical) then set

$\bar{\lambda}^* = \bar{\lambda}^-$. Find $\bar{\mathbf{S}}^u$ and $\bar{\mathbf{S}}^+$ by substituting $\lambda = \bar{\lambda}^*$ in (35)-(36).

Stop

The complexity of Algorithm 2 is $O(N)$ (See Appendix A).

According to (44) we have $b_i^* = \{b_i^*(\bar{\lambda}^*)\}$ or $b_i^* = \{b_i^*(\bar{\lambda}^*)\} + 1$. Thus, \mathbf{S}^+ and \mathbf{S}^u can be found from $\mathbf{S}^+ = \bar{\mathbf{S}}^+ + \Delta\mathbf{S}^+$ and $\mathbf{S}^u = \bar{\mathbf{S}}^u + \Delta\mathbf{S}^u$ where $\Delta\mathbf{S}^+$ is the set of subcarriers for which $\{b_i^*(\bar{\lambda}^*)\} + 1 = b_i^* = 1$ and $\Delta\mathbf{S}^u$ is the set of subcarriers for which $\{b_i^*(\bar{\lambda}^*)\} + 1 = b_i^* = u_i$. Assume that $d = B - \{B^*(\bar{\lambda}^*)\}$. Since $\{b_i^*(\lambda)\}$ s and $\{B^*(\lambda)\}$ are non-increasing with λ , the subcarriers for which $\{b_i^*(\bar{\lambda}^*)\} + 1 = b_i^*$ are those d subcarriers with larger $\lambda_{i, \{b_i^*(\bar{\lambda}^*)+1\}}^c$ s.

Algorithm 3: Finding \mathbf{S}^+ and \mathbf{S}^u

Step 1- Using Algorithm 2 find $\bar{\mathbf{S}}^+$, $\bar{\mathbf{S}}^u$ and $\bar{\lambda}^*$.

Step 2- Find $\{b_i^*(\bar{\lambda}^*)\}$ s for the subcarriers in $\bar{\mathbf{S}}'$ from (18) and find $\{B^*(\bar{\lambda}^*)\}$ from (21). If $\{B^*(\bar{\lambda}^*)\} = B$ then we have $\lambda^* = \bar{\lambda}^*$; set $\mathbf{S}^+ = \bar{\mathbf{S}}^+$ and $\mathbf{S}^u = \bar{\mathbf{S}}^u$. Otherwise set $d = B - \{B^*(\bar{\lambda}^*)\}$ and find

$$\bar{\Lambda} = \Phi\left(d, \left\{ \lambda_{i, \{b_i^*(\bar{\lambda}^*)+1\}}^c \right\}_{i \in \mathbf{S}}\right)$$

$$\Delta\mathbf{S}^+ = \left\{ i : \lambda_{i, \{b_i^*(\bar{\lambda}^*)+1\}}^c \in \bar{\Lambda} \text{ and } \{b_i^*(\bar{\lambda}^*)\} + 1 = 1 \right\}$$

$$\Delta\mathbf{S}^u = \left\{ i : \lambda_{i, \{b_i^*(\bar{\lambda}^*)+1\}}^c \in \bar{\Lambda} \text{ and } \{b_i^*(\bar{\lambda}^*)\} + 1 = u_i \right\}$$

then set $\mathbf{S}^+ = \bar{\mathbf{S}}^+ + \Delta\mathbf{S}^+$ and $\mathbf{S}^u = \bar{\mathbf{S}}^u + \Delta\mathbf{S}^u$.

Stop

The worst case complexity of Algorithm 3 is $O(N)$ (See Appendix A). According to Theorem 3 and Lemma 3 the solution of ONIBL problem is obtained as

$$b_i^* = \begin{cases} 0 & \text{if } i \notin \mathbf{S}^+ \\ u_i & \text{if } i \in \mathbf{S}'' \\ \tilde{b}^+ - \lfloor \log_2(C_i) \rfloor + 1 & \text{if } i \in \mathbf{S}' \text{ and } \rho_i \in \tilde{\mathbf{P}}^+ \\ \tilde{b}^+ - \lfloor \log_2(C_i) \rfloor & \text{otherwise} \end{cases} \quad (46)$$

where

$$B_0^+ = \sum_{i \in \mathbf{S}'} \lfloor \log_2(C_i) \rfloor, \quad B'' = \sum_{i \in \mathbf{S}''} u_i, \quad \tilde{b}^+ = \left\lfloor \frac{B - B'' + B_0^+}{N'} \right\rfloor,$$

$$\tilde{\mathbf{B}}^+ = B - B'' + B_0^+ - N' \tilde{b}^+, \quad \tilde{\mathbf{P}}^+ = \Phi'(\tilde{\mathbf{B}}^+, \{\rho_i : i \in \mathbf{S}'\})$$

Since \mathbf{S}^+ and \mathbf{S}'' can be found by using Algorithm 3 which is $O(N)$, the complexity of computing the analytical solution (46) for all subcarriers is altogether $O(N)$. Thus, the proposed method has lower computational complexity compared to greedy and EBF/EBR algorithms.

a) Comparison with the existing algorithms

To explain what has happened in the proposed analytical method, the main differences between the proposed algorithm, greedy algorithm and EBF/EBA algorithm are explained below:

- a) In the proposed method, to find a group of subcarriers, the selection algorithm with the complexity of $O(N)$ is used (in Algorithm 2), while the EBR/EBF algorithm for similar purpose uses the sorting algorithm whose complexity is $O(N \log N)$.
- b) The greedy algorithm allocates only one bit to one subcarrier at each iteration, while in EBR/EBF algorithm one bit is allocated to a group of subcarriers at each iteration, and in the proposed algorithm, after finding \mathbf{S}^+ , all the bits are allocated in one iteration using the analytical solution. Thus, the proposed algorithm has fewer iterations. However, in the proposed method first we need to find \mathbf{S}^+ . Fortunately, \mathbf{S}^+ can be found by using a low computational complexity $O(N)$ algorithm (Algorithm 3).
- c) In greedy and EBR/EBF algorithms, at each iteration a search *among all subcarriers* is required for finding the subcarriers with lower transmission power, while in the proposed algorithm, in the stage of finding \mathbf{S}^+ , using the bisection algorithm the set of subcarriers is bisected at each iteration and the complexity of search decreases to half in the next iterations so that the total complexity would be $O(N)$ (as shown in proof of algorithms 2 and 3 in Appendix A).
- d) Because of one-by-one bit loading in greedy algorithm, its complexity grows with B . Similarly, due to group-by-group bit loading the complexity of EBF/EBR algorithm grows with b_{max} [9]. While in the proposed method by using the analytical solution, the complexity does not grow with B and b_{max} . It should be noted that in [9], first b_{max} is considered as a constant and the complexity of EBF/EBR algorithm is considered as $O(N \log_2 N)$ but in [9, Table I] it is shown that the complexity of EBF/EBR algorithm depends on b_{max} too.

5. SIMULATION RESULTS

A computer simulation is carried out to compare the analytical solution (46) with the greedy algorithm [1-3] and EBF/EBR algorithm proposed in [9]. The results confirm that the proposed analytical solution (46) is optimal and the solutions obtained from (46) are the same as those obtained from the existing optimal algorithms. Some of the results for different values of C_1 to C_N , B and N are shown in Table 1.

Table 1. Solutions of ONIBL problem obtained from analytical relation (46), EBF/EBR algorithm and the greedy algorithm in some cases. The results are the same for all three methods

N	B	C_1 to C_N	u_1 to u_N	P_{opt} (dB)	b_1^* to b_N^*
16	32	5.7, 4.7, 13.3, 15.2, 9.8, 14.0, 15.4, 10.1, 12.5, 6.3, 7.5, 1.0, 12.6, 5.5, 13.3, 15.5	-	26.08	3, 3, 1, 1, 2, 1, 1, 2, 2, 2, 2, 5, 2, 3, 1, 1
16	96	6.3, 2.0, 5.0, 1.0, 2.7, 6.0, 5.0, 5.0, 6.1, 2.1, 4.7, 2.1, 6.8, 5.6, 5.9, 5.3	8	36.13	5, 7, 6, 8, 7, 5, 6, 6, 5, 7, 6, 7, 5, 5, 5, 6
32	128	26.0, 13.3, 4.3, 5.2, 26.7, 1.0, 17.8, 27.0, 31.0, 15.1, 2.3, 17.1, 6.4, 9.8, 31.9, 5.4, 25.4, 11.2, 15.5, 3.3, 2.0, 28.2, 2.3, 28.9, 12.9, 14.4, 11.5, 1.1, 23.7, 11.6, 21.0, 25.3	-	36.97	3, 4, 5, 5, 3, 7, 3, 3, 2, 3, 6, 3, 5, 4, 2, 5, 3, 4, 3, 6, 6, 3, 6, 2, 4, 4, 4, 7, 3, 4, 3, 3
32	256	385.9, 276.9, 462.8, 43.3, 16.1, 247.0, 81.8, 460.5, 1.0, 54.8, 240.3, 134.0, 545.0, 211.7, 280.0, 152.4, 328.4, 296.4, 557.8, 376.5, 482.3, 49.7, 241.0, 23.6, 343.1, 416.8, 389.0, 34.8, 385.5, 175.0, 198.5, 414.0	10	61.83	7, 8, 7, 10, 10, 8, 9, 7, 10, 10, 8, 9, 7, 8, 7, 8, 7, 7, 6, 7, 7, 10, 8, 10, 7, 7, 7, 10, 7, 8, 8, 7

Another computer simulation is carried out to compare the running time required for computing (46) with the running time of greedy and EBF/EBR algorithms. The specifications of the computer used for running all algorithms are: CPU: Intel B970, dual core, 2.3 GHz; RAM: 4 GB; OS: Windows 7, 64-bit. The results for $N = 32$ to 1024 with random values of C_i s and u_i s are shown in Figs. 5 and 6. The results confirm that the proposed optimal bit loading method is $O(N)$ and is faster than greedy and EBF/EBR algorithms.

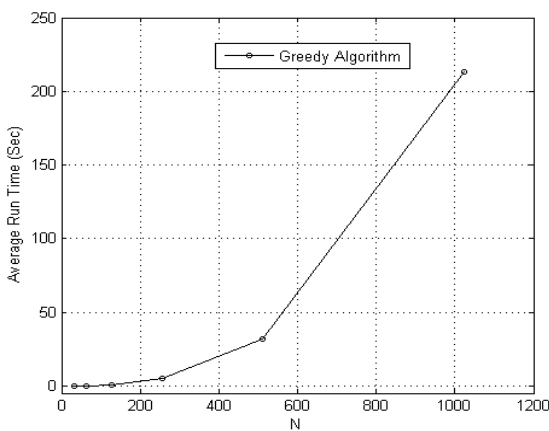


Fig. 5. Average running time of greedy algorithm for $N=32$ to 1024

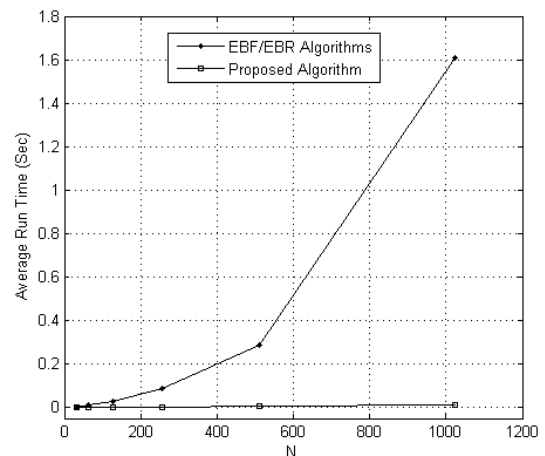


Fig. 6. Average running time of EBF/EBR algorithm and proposed method for $N=32$ to 1024

Another computer simulation is performed to evaluate the required number of operations (including comparison, division, addition, shift and function operations) of all three methods for different values of N . The results of average number of required operations in 100 runs with $N = 32$ to 1024, $B = 2N$ and $u_i = b_{max} = 0.5N$, $i = 1, 2, \dots, N$ are shown in Fig. 7.

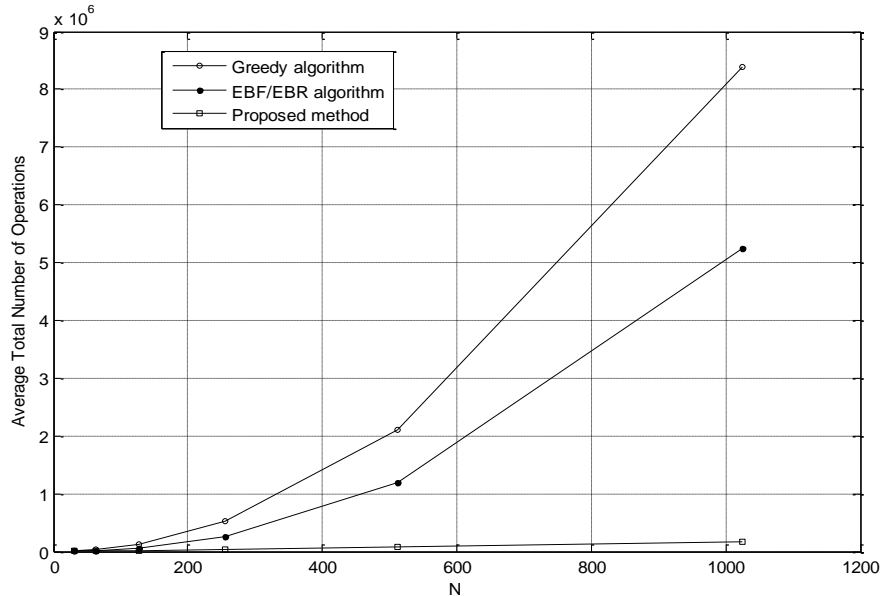


Fig. 7. The average number of required operations in greedy algorithm, EBF/EBR algorithm and proposed method for $N = 32$ to 1024, $B = 2N$ and $b_{max} = 0.5N$.

The average is taken over 100 runs

The simulation results are compatible with the complexity analysis in [9]. These results also confirm the mathematical proofs which showed that the proposed method is $O(N)$ and has lower computational complexity compared to greedy and EBF/EBR algorithms.

6. CONCLUSION

In this paper we derived an analytical solution for optimal integer bit loading in OFDM systems. At first, we obtained an analytical solution in terms of the Lagrange multiplier, λ . Then we derived an analytical solution for integer bit loading. Finally, we modified the solution for nonnegative integer bit loading with upper constraints on the number of bits of each channel. The complexity of computing the analytical solution is $O(N)$, which is less than the complexity of existing algorithms. In addition to the mathematical proofs, computer simulations confirmed that not only is the proposed analytical solution optimal but also it is faster than the greedy and EBF/EBR algorithms.

REFERENCES

1. Baccarelli, E. & Biagi, M. (2004). Optimal integer bit-loading for multicarrier adsl systems subject to spectral-compatibility limits. *Signal processing*, Vol. 84, No. 4, pp. 729–741.
2. Fasano, A. (2003). On the optimal discrete bit loading for multicarrier systems with constraints. *Vehicular Technology Conference, VTC 2003-Spring*. The 57th IEEE Semiannual. Vol. 2. IEEE, pp. 915–919.
3. Fasano, A., Di Blasio, G., Baccarelli, E. & Biagi, M. (2002). Optimal discrete bit loading for dmt based constrained multicarrier systems. In: *Information Theory, 2002. Proceedings. 2002 IEEE International Symposium on*. IEEE, p. 243.

4. Fischer, R. F. & Huber, J. B. (1996). A new loading algorithm for discrete multitone transmission. In: Global Telecommunications Conference. *GLOBECOM'96. 'Communications: The Key to Global Prosperity*, Vol. 1. IEEE, pp. 724–728.
5. Gong, X., Liu, Y. & Guo, L. (2014). Joint optimization of bit and power loading for power efficient OFDM-PON. *Optical Communications and Networks (ICOON), 13th International Conference on*, Vol., No., pp.1-10.
6. Wang, A. & Li, S. (2007). A flexible bit allocation algorithm based on adaptive method for ofdm channels. In: *Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on. IEEE*, pp. 291–294.
7. Sun, Q., Tolli, A., Juntti, M. & Mao, J. (2014). Optimal energy efficient bit and power loading for multicarrier systems. *Communications Letters, IEEE*, Vol.18, No.7, pp. 1194-1197.
8. Piazzo, L. (2011). Optimal fast algorithm for power and bit allocation in ofdm systems. *Vehicular Technology, IEEE Transactions*, Vol. 60, No. 3, pp. 1263–1265.
9. Wang, D., Cao, Y. & Zheng, L. (2010). Efficient two-stage discrete bit loading algorithms for ofdm systems. *Vehicular Technology, IEEE Transactions*, Vol. 59, No. 7, pp. 3407–3416.
10. Baccarelli, E., Fasano, A. & Biagi, M. (2002). Novel efficient bit-loading algorithms for peak-energy-limited adsl-type multicarrier systems. *Signal Processing, IEEE Transactions*, Vol. 50, No. 5, pp. 1237–1247.
11. Cioffi, J. M., Dudevoir, G. P., Veday Eyuboglu, M. & Forney, Jr, G. D. (1995). Mmse decision-feedback equalizers and coding. i. equalization results. *Communications, IEEE Transactions*, Vol. 43, No. 10, pp. 2582–2594.
12. Ko, H., Lee, K., Oh, S. & Kim, C. (2009). Fast optimal discrete bit-loading algorithms for ofdm-based systems. In: *Computer Communications and Networks, 2009. ICCCN 2009. Proceedings of 18th International Conference on. IEEE*, pp. 1–6.
13. Tao, Z. & Rui, Z. (2010). A novel bit and power allocation scheme for ofdm system over the low voltage power line. In: *Natural Computation (ICNC), Sixth International Conference*, Vol. 5. IEEE, pp. 2540–2544.
14. Zhen, L., Weihua, W., Wenan, Z. & Junde, S. (2003). A simple bit and power allocation algorithm for ofdm systems. In: *Communication Technology Proceedings, 2003. ICCT 2003. International Conference*, Vol. 2. IEEE, pp. 1169–1172.
15. Chow, P. S., Cioffi, J. M. & Bingham, J. (1995). A practical discrete multitone transceiver loading algorithm for data transmission over spectrally shaped channels. *IEEE Transactions on communications*, Vol. 43, No. 234, pp. 773–775.
16. Starr, T., Cioffi, J. M. & Silverman, P. J. (1999). *Understanding digital subscriber line technology*. Prentice Hall PTR.
17. Jang, J. & Lee, K. B. (2003). Transmit power adaptation for multiuser OFDM systems. *Selected Areas in Communications, IEEE Journal*, Vol. 21, No. 2, pp. 171-178.
18. Papandreou, N. & Antonakopoulos, T. (2005). A new computationally efficient discrete bit-loading algorithm for dmt applications. *Communications, IEEE Transactions*, Vol. 53, No. 5, pp. 785–789.
19. Hatam, M. & Masnadi-Shirazi, M. A. (2008). Analytical discrete optimization. *Iranian Journal of Science and Technology*, Vol. 32 (B), p. 249.
20. Everett, H. (1963). Generalized lagrange multiplier method for solving problems of optimum allocation of resources. *Operations research*, Vol. 11, No. 3, pp. 399–417.
21. Blum, M., Floyd, R. W., Pratt, V., Rivest, R. L. & Tarjan, R. E. (1973). Time bounds for selection. *Journal of Computer and System Sciences*, Vol. 7, No. 4, pp. 448–461.
22. Dardari, D. (2004). Ordered subcarrier selection algorithm for ofdm based high-speed w lans. *Wireless Communications, IEEE Transactions*, Vol. 3, No. 5, pp. 1452–1458.
23. Floyd, R. W. & Rivest, R. L. (1975). Expected time bounds for selection. *Communications of the ACM*, Vol. 18, No. 3, pp. 165–172.

APPENDIX A

Proof of Theorems, Lemmas and Complexity of Algorithms

Proof of Theorem 1

Assume that n^* is a local minimum of $f[n]$. We have

$$\begin{aligned} f[n^*] \leq f[n^* - 1] &\Rightarrow f_r(n^*) \leq f_r(n^* - 1) \\ &\Rightarrow f_r(n^*) - f_r(n^* - 1) \leq 0 \end{aligned} \tag{A1}$$

Also we have

$$\begin{aligned} f[n^*] \leq f[n^* + 1] &\Rightarrow f_r(n^*) \leq f_r(n^* + 1) \\ &\Rightarrow f_r(n^* + 1) - f_r(n^*) \geq 0 \end{aligned} \tag{A2}$$

Since $f_r(x)$ is a continuous function on the interval $I_r = [a, b]$, $f_r(x - 1)$ is also a continuous function on the interval $[a + 1, b + 1]$ and therefore $g(x) = f_r(x) - f_r(x - 1)$ is a continuous function on $[a + 1, b]$. Assume that $I = \{n_1, n_1 + 1, \dots, n_2\} \subset I_r$. According to definition of local minima (subsection 3-a) we have $n^* \in \{n_1 + 1, n_1 + 1, \dots, n_2 - 1\} \subset [a + 1, b - 1]$ and consequently $[n^*, n^* + 1] \subset [a + 1, b]$. According to (A1) and (A2) we have $g(n^*) \leq 0$ and $g(n^* + 1) \geq 0$ and since $g(x)$ is a continuous function, according to Bolzano's theorem it has at least one root in the interval $[n^*, n^* + 1]$. Thus, the equation $f_r(x) - f_r(x - 1) = 0$ has at least one solution in the interval $[n^*, n^* + 1]$. If the solution(s) is (are) in the interval $[n^*, n^* + 1]$, the floor of the solution(s) would be n^* . If the solution is equal to $n^* + 1$, since $n^* + 1$ is integer, $n^* + 1$ and $(n^* + 1) - 1 = n^*$ would be in the set \mathbf{A} . Thus, in either case the set \mathbf{A} contains the local minimum n^* . Similarly, it can be shown that if n^* is a local maximum of $f[n]$, the set \mathbf{A} contains n^* [19].

Proof of Theorem 2

Since $\mathbf{x}^*(\lambda)$ is the minimum of $\{G(\mathbf{x}) + \lambda C(\mathbf{x})\}$ in \mathbf{D} we have:

$$\begin{aligned} G(\mathbf{x}^*(\lambda)) + \lambda C(\mathbf{x}^*(\lambda)) &\leq G(\mathbf{x}) + \lambda C(\mathbf{x}), \forall \mathbf{x} \in \mathbf{D} \\ G(\mathbf{x}^*(\lambda)) - G(\mathbf{x}) &\leq \lambda C(\mathbf{x}) - \lambda C(\mathbf{x}^*(\lambda)), \forall \mathbf{x} \in \mathbf{D} \end{aligned}$$

If $C(\mathbf{x}^*(\lambda)) = c$ then we have

$$G(\mathbf{x}^*(\lambda)) - G(\mathbf{x}) \leq \lambda(C(\mathbf{x}) - c), \forall \mathbf{x} \in \mathbf{D}$$

For the vectors \mathbf{x} that satisfy the constraint of the problem, $C(\mathbf{x}) = c$, we have

$$\begin{aligned} G(\mathbf{x}^*(\lambda)) - G(\mathbf{x}) &\leq 0, \forall \mathbf{x} \in \mathbf{D} \\ G(\mathbf{x}^*(\lambda)) &\leq G(\mathbf{x}), \forall \mathbf{x} \in \mathbf{D} \end{aligned}$$

Thus $\mathbf{x}^*(\lambda)$ is the solution of the problem (6).

Proof of Strong Duality

According to Theorem 2, for proving strong duality we should show that there always exists a λ at which we have $\{B^*(\lambda)\} = B$.

For OIBL problem, according to (17), $\{b_i^*(\lambda)\}$ contains all the integer values in \mathbb{Z} . Since $\{b_i^*(\lambda)\}$ s are non-increasing and $\lim_{\lambda \rightarrow -\infty} \{b_i^*(\lambda)\} = \infty$ and $\lim_{\lambda \rightarrow \infty} \{b_i^*(\lambda)\} = -\infty$ then $\{B^*(\lambda)\} = \sum_{i=1}^N \{b_i^*(\lambda)\}$ contains all the integer values including B and $\{B^*(\lambda)\} = B$ always can be held.

For ONIBL problem, according to (18), $\{b_i^*(\lambda)\}$ contains all the integer values in the interval $[0, u_i]$. Since $\{b_i^*(\lambda)\}$ s are non-increasing and $\lim_{\lambda \rightarrow -\infty} \{b_i^*(\lambda)\} = u_i$ and $\lim_{\lambda \rightarrow \infty} \{b_i^*(\lambda)\} = 0$ then $\{B^*(\lambda)\} = \sum_{i=1}^N \{b_i^*(\lambda)\}$

contains all the integer values in the interval $[0, \sum_{i=1}^N u_i]$ including B (according to the condition $\sum_{i=1}^N u_i \geq B$) and $\{B^*(\lambda)\} = B$ always can be held.

Therefore, according to Theorem 2, the strong duality property is satisfied and the solution of Lagrange dual problem is the same as that of the main problem.

Complexity of Algorithm 1

In step 2 of the algorithm for $f[\bar{n}] > 0$ the value n_1 is set to $\bar{n} + 1$ and for $f[\bar{n}] < 0$ the value n_2 is set to $\bar{n} - 1$. Thus, in i^{th} iteration we would have

$$N_i \leq \frac{N_{i-1}}{2} \leq \frac{N}{2^i}$$

Where N_i is the value of $n_2 - n_1$ at the i^{th} iteration. Assume that I is the total number of iterations then

$$\frac{N}{2^I} \leq 1$$

$$I = \lceil \log_2 N \rceil + 1$$

Thus, the algorithm is $O(\log_2 N)$.

Proof of Lemma 1

Assume that $\lambda_{k,j_1}^c \in [\lambda_{i,j+1}^c, \lambda_{i,j}^c]$ then

$$-C_i 2^j \leq -C_k 2^{j_1} < -C_i 2^{j-1} \quad (\text{A3})$$

Then for each integer $j_2 \neq j_1$ we have

$$j_2 \geq j_1 + 1 \text{ or } j_2 \leq j_1 - 1$$

$$-C_k 2^{j_2} \leq -C_k 2^{j_1+1} \text{ or } -C_k 2^{j_2} \geq -C_k 2^{j_1-1} \quad (\text{A4})$$

According to (A3) we have

$$-C_i 2^{j+1} \leq -C_k 2^{j_1+1} < -C_i 2^j \quad (\text{A5})$$

and

$$-C_i 2^{j-1} \leq -C_k 2^{j_1-1} < -C_i 2^{j-2} \quad (\text{A6})$$

According to (A4)-(A6) $\lambda_{k,j_2}^c \notin [\lambda_{i,j+1}^c, \lambda_{i,j}^c]$. Thus, $\{b_k^*(\lambda)\}$ has at most one critical λ in the interval $[\lambda_{i,j+1}^c, \lambda_{i,j}^c]$ (here λ_{k,j_1}^c).

Proof of Lemma 2

To solve the ONIBL problem we can find the intersection of $\{B^*(\lambda)\}$ with the horizontal line $f(\lambda) = B$. Since $\{B^*(\lambda)\}$ is non-increasing with λ , we may encounter the following cases:

1) The intersection of $\{B^*(\lambda)\}$ with $f(\lambda) = B$ is a singular point. In this case the solutions occur at a critical λ as λ_s and two or more number of $\{b_i^*(\lambda)\}$ s have two values at λ_s . Assume that m number of $\{b_i^*(\lambda)\}$ s have jump at λ_s ($m \geq 2$) and let $l = B^*(\lambda_s)_U - B$. In order to have $\{B^*(\lambda)\} = B$, since $l \geq 0$, for l number of $\{b_i^*(\lambda)\}$ s which have two values at λ_s we must select the lower values and for the remaining $\{b_i^*(\lambda)\}$ s select the upper values. Thus, in this case we have $\binom{m}{l}$ solutions for ONIBL problem. Also $\{B^*(\lambda)\}$ would have $m - 1$ singular points and two nonsingular steps at λ_s . (see Fig. 4).

2) Intersection of $\{B^*(\lambda)\}$ with $f(\lambda) = B$ is a nonsingular step (defined in section 4) that contains infinite values for λ . Since we have no jump at a nonsingular step, the values of $\{b_i^*(\lambda)\}$ s are constant at a nonsingular step. Thus, in this case the ONIBL problem has a unique solution. \square

Proof of Lemma 3

In the ONIBL problem we should find a λ such that

$$\begin{aligned} \sum_{i=1}^N \{b_i^*(\lambda)\} &= B \\ \sum_{i:0 < \{b_i^*(\lambda)\} < u_i} \{b_i^*(\lambda)\} + \underbrace{\sum_{i:\{b_i^*(\lambda)\}=0} \{b_i^*(\lambda)\}}_{=0} + \sum_{i:\{b_i^*(\lambda)\}=u_i} \{b_i^*(\lambda)\} &= B \\ \sum_{i:0 < \{b_i^*(\lambda)\} < u_i} \{b_i^*(\lambda)\} &= B - \sum_{i:\{b_i^*(\lambda)\}=u_i} u_i \end{aligned}$$

Substituting $\lambda = \lambda^*$ yields

$$\sum_{i \in \mathbf{S}'} \{b_i^*(\lambda^*)\} = B - \sum_{i \in \mathbf{S}''} u_i$$

We should find a λ^* that satisfies the above relation. Thus, the ONIBL problem is an OIBL problem for the subcarriers in \mathbf{S}' and symbol size $B - \sum_{i \in \mathbf{S}''} u_i$.

Proof of Lemma 4

According to (18), (24) and Remark 3 we have

a)

$$\lambda^* > \lambda_{i,1}^c \Leftrightarrow \{b_i^*(\lambda^*)\} < 1 \Leftrightarrow b_i^* = 0$$

b)

$$\lambda^* < \lambda_{i,u_i}^c \Leftrightarrow \{b_i^*(\lambda^*)\} \geq u_i \Leftrightarrow b_i^* = u_i$$

Proof of Theorem 3

In (27) b_i^* can be rewritten as

$$\begin{aligned} b_i^* &= \left\lfloor \frac{B}{N} - \frac{1}{N} \sum_{n=1}^N [\log_2(C_i) - \log_2(C_n)] \right\rfloor \\ &= \left\lfloor \frac{B}{N} + \frac{B_0}{N} - [\log_2(C_i)] - \frac{1}{N} \sum_{n=1}^N [\rho_i - \rho_n] \right\rfloor \end{aligned} \tag{A7}$$

where

$$B_0 = \sum_{n=1}^N [\log_2(C_n)] \tag{A8}$$

$$\rho_n = \log_2(C_n) - [\log_2(C_n)], \quad n = 1, 2, \dots, N \tag{A9}$$

Since $\rho_n \in [0, 1)$, $\forall n$ then $(\rho_i - \rho_n) \in (-1, 1)$ and therefore $[\rho_i - \rho_n] \in \{-1, 0\}$. Thus

$$0 \leq -\frac{1}{N} \sum_{n=1}^N [\rho_i - \rho_n] < 1$$

define

$$B_{i0} = -\sum_{n=1}^N [\rho_i - \rho_n] \tag{A10}$$

$$\tilde{b} = \left\lfloor \frac{B + B_0}{N} \right\rfloor \tag{A11}$$

$$\tilde{B} = B + B_0 - N\tilde{b} \tag{A12}$$

Substituting (A10)-(A12) in (A7) yields

$$\begin{aligned} b_i^* &= \left\lfloor \tilde{b} + \frac{\tilde{B} + B_{i0}}{N} - [\log_2(C_i)] \right\rfloor \\ &= \tilde{b} + \left\lfloor \frac{\tilde{B} + B_{i0}}{N} \right\rfloor - [\log_2(C_i)] \end{aligned}$$

Since \tilde{B} is the reminder of dividing $B + B_0$ to N we have $0 \leq \tilde{B} < N$. Also, we have $0 \leq B_{i0} < N$ and hence

$$0 \leq \frac{\tilde{B} + B_{i0}}{N} < 2 \quad (\text{A13})$$

Thus, according to (A12) and (A13):

$$b_i^* = \begin{cases} \tilde{b} - \lfloor \log_2(C_i) \rfloor + 1 & \text{if } B_{i0} \geq N - \tilde{B} \\ \tilde{b} - \lfloor \log_2(C_i) \rfloor & \text{otherwise} \end{cases} \quad (\text{A14})$$

On the other hand, we have

$$\lfloor \rho_i - \rho_n \rfloor = \begin{cases} -1 & \text{if } \rho_n > \rho_i \\ 0 & \text{if } \rho_n \leq \rho_i \end{cases} \quad (\text{A15})$$

We can conclude from (A9), (A10) and (A15) that

$$B_{i0} = \mathfrak{R}(\rho_i, \{\rho_n\}_{n=1}^N) - 1$$

Define

$$\tilde{\mathbf{P}} = \Phi'(\tilde{B}, \{\rho_n\}_{n=1}^N)$$

The operators Φ' and \mathfrak{R} are defined in subsection 3-c. We have

$$B_{i0} \geq N - \tilde{B} \Leftrightarrow \mathfrak{R}(\rho_i, \{\rho_n\}_{n=1}^N) - 1 \geq N - \tilde{B} \Leftrightarrow \rho_i \in \tilde{\mathbf{P}}$$

Thus, (A14) can be written as

$$b_i^* = \begin{cases} \tilde{b} - \lfloor \log_2(C_i) \rfloor + 1 & \text{if } \rho_i \in \tilde{\mathbf{P}} \\ \tilde{b} - \lfloor \log_2(C_i) \rfloor & \text{otherwise} \end{cases}$$

In the case that some of b_i s have two optimal values, according to Lemma 2 the corresponding $\{b_i^*(\lambda)\}$ s have equal critical λ s which are equal to the optimal λ . Assume that b_i and b_k have two optimal values then we have

$$\begin{aligned} \lambda_{i,b_i^*}^c &= \lambda_{k,b_k^*}^c \\ C_i 2^{b_i^*-1} &= C_k 2^{b_k^*-1} \end{aligned}$$

Taking \log_2 yields

$$\log_2(C_i) + b_i^* - 1 = \log_2(C_k) + b_k^* - 1$$

Substituting b_i^* and b_k^* from (27) we conclude $\rho_i = \rho_k$ or $\rho_i = \rho_k + 1$ or $\rho_i = \rho_k - 1$.

Since $\rho_n \in [0,1)$, $\forall n$, only the first equation is acceptable and we have $\rho_i = \rho_k$. Let m be the number of b_i s that have two optimal values (have equal ρ_i s) then the difference between sum of the upper values of b_i^* and B is

$$\begin{aligned} l &= \left(\sum_{i=1}^N (\tilde{b} - \lfloor \log_2(C_i) \rfloor + 1) \right) - B \\ l &= (N\tilde{b} + N - B_0) - B \end{aligned}$$

Thus, in order to have $\sum_{i=1}^N b_i^* = B$ for l number of b_i^* s we should select the lower value in (28) and we have $\binom{m}{l}$ combinations for the solution of problem.

Proof of Lemma 5

- a) It is obvious from (18)-(20).
- b) According to (20):

$$\begin{aligned} \bar{B}(\lambda) &= \sum_{i=1}^N \bar{b}_i(\lambda) \\ &= \sum_{i \in \mathcal{S}'(\lambda)} \bar{b}_i(\lambda) + \sum_{i \in \mathcal{S}^u(\lambda)} u_i \end{aligned}$$

Substituting $\bar{b}_i(\lambda)$ from (19) yields

$$\bar{B}(\lambda) = N'(\lambda) \log_2(-2\lambda) - B'(\lambda) + B^u(\lambda)$$

- c) According to (42):

$$\bar{B}(\bar{\lambda}^*) = N'(\bar{\lambda}^*) \log_2(-2\bar{\lambda}^*) - B'(\bar{\lambda}^*) + B^u(\bar{\lambda}^*)$$

If $\bar{\lambda}^*$ is non-critical then we have $\bar{B}(\bar{\lambda}^*) = B$ and hence

$$\begin{aligned} B &= \bar{N}' \log_2(-2\bar{\lambda}^*) - \bar{B}' + \bar{B}^u \\ \bar{\lambda}^* &= -2^{(B + \bar{B}' - \bar{B}^u) / \bar{N}' - 1} \end{aligned}$$

- d) According to (41):

$$\begin{aligned} \bar{B}(\bar{\lambda}^*) - N'(\bar{\lambda}^*) &\leq \{B^*(\bar{\lambda}^*)\} \leq \bar{B}(\bar{\lambda}^*) \\ B - \bar{N}' &\leq \{B^*(\bar{\lambda}^*)\} \leq B \end{aligned} \tag{A16}$$

Since for the subcarriers in $\bar{\mathcal{S}}'$ we have $0 < b_i^*(\bar{\lambda}^*) < u_i$, all these \bar{N}' subcarriers have at least one critical λ smaller than $\bar{\lambda}^*$. Furthermore, since $\bar{N}' \leq N$ according to (A16) and Lemma 1 each $\{b_i^*(\lambda)\}$ in the interval $[\lambda^*, \bar{\lambda}^*)$ has at most one critical λ (otherwise at least one of $\{b_i^*(\lambda)\}$ s has more than one critical λ between two successive critical λ s of another $\{b_i^*(\lambda)\}$ which is contradiction to Lemma 1). Thus, since at each critical λ the value of $\{b_i^*(\lambda)\}$ changes 1 unit we have

$$\begin{aligned} b_i^*(\lambda^*) - 1 &\leq b_i^*(\bar{\lambda}^*) \leq b_i^*(\lambda^*), \forall i \\ b_i^* - 1 &\leq b_i^*(\bar{\lambda}^*) \leq b_i^*, \forall i \end{aligned}$$

Complexity of Algorithm 2

This algorithm has at most $\lfloor \log_2(N) \rfloor + 1$ iterations and at each iteration the sets $\bar{\Lambda}$, $\Delta \bar{\mathcal{S}}^u$, $\Delta \bar{\mathcal{S}}^+$, $\Delta \bar{\mathcal{S}}'$, $\Delta \bar{B}^u$, and $\Delta \bar{B}'$ are bisected. Thus, according to the complexity of selection algorithm, the computational complexity of l^{th} iteration is $O(N / 2^l)$. Therefore, the order of the total complexity of the algorithm is

$$\sum_{l=0}^{\lfloor \log_2(N) \rfloor + 1} \frac{N}{2^l} = N \frac{1 - (1/2)^{\lfloor \log_2(N) \rfloor + 2}}{1 - (1/2)} \leq 2N$$

Complexity of Algorithm 3

The worst case complexity of computing $\{B^*(\bar{\lambda}^*)\}$, $\bar{\Lambda}$, $\Delta \mathcal{S}^+$, $\Delta \mathcal{S}^u$ and complexity of Algorithm 2 is $O(N)$ and hence Algorithm 3 is also $O(N)$.