

Efficient Multi-User Parallel Greedy Bit-Loading Algorithm with Fairness Control For DMT Systems

Cajetan M. Akujuobi and Jie Shen

*The Center of Excellence for Communication Systems Technology Research (CECSTR),
Prairie View A&M University, Prairie View, Texas 77446
USA*

1. Introduction

This chapter addresses the multi-user bit-loading algorithm for discrete multitone (DMT) modulation in digital subscriber line (DSL) systems. The widely deployed asymmetric digital subscriber line (ADSL) provides the high bit rate data transmission as well as plain old telephone service (POTS) on a single twisted-pair at the same time. DMT, the core of DSL systems, divides the frequency-selective channel into large number of narrow subchannels. If the number is large enough, each subchannel becomes flat in frequency response, although the responses may differ a lot among subchannels. One of the advantages of DMT is that the power spectral density (PSD) and bits allocated to each subchannel could be chosen according to the subchannel signal-to-noise ratio (SNR) in order to obtain the optimal performance (e.g. maximum data rate, or minimum power consumption). This process is called bit loading and is a critical issue in the design of DMT systems.

In the early days of DMT development, bit loading was studied only in single-user case, where only one pair of modems (transmitter and receiver) was considered. Compared with traditional telephone service, DSL systems always work on high frequency range, which causes the crosstalk interference among the twisted pairs in the same cable noticeable. The SNR of a subchannel is related not only with the PSD of its own transmitter, but also with the PSD of all other transmitters in the same cable that act as disturbers. The bit-loading algorithms need to be extended to multi-user scenario to obtain the global optimum performance among all users. The optimal algorithm for discrete multi-user bit loading is a natural extension of single-user greedy algorithm. A matrix of cost is calculated, with elements that represent the power increment to transmit additional bits for each subchannel. Then, the subchannel with minimum cost is found, and additional bits are assigned to it. The process continues until all subchannels are filled. A drawback of the multi-user greedy bit loading is the computation complexity. For a single iteration of the algorithm, only one subchannel on one user who has the minimum cost is selected to get additional bits.

The objective of this chapter is to propose an efficient greedy bit-loading algorithm for multi-user DMT systems. An improved parallel bit-loading algorithm for multi-user DMT will be discussed. The new algorithm is based on multi-user greedy bit loading. In a single

Source: Advances in Greedy Algorithms, Book edited by: Witold Bednorz,
ISBN 978-953-7619-27-5, pp. 586, November 2008, I-Tech, Vienna, Austria

iteration, the bits were allocated to multiple users on the subchannels with same frequency. Thus, the number of iterations to allocate bits to all users decreased significantly. The adjustable cost elastic coefficient defined the range of power cost. In the bit-loading iteration, all subchannels for different users that have the power cost within the range have the chance to get additional bits assigned to them.

As a consequence of the greedy bit-loading algorithm, the user with better channel condition, which means it has smaller cost to transmit additional bits, will have more chance to get bits assigned to it until it meets the target data rate or exceeds the power budget. The user with worse channel condition is sacrificed in order to gain the maximum total data rate. However, in most real networks, users in the same cable are of equal priority. Their service quality is supposed to be as equal as possible. Fairness should be considered in the design of bit-loading algorithm. This chapter studied the possibility to improve the fairness among users in the same cable. We proposed a fairness control method in the bit-loading algorithm. A fairness coefficient is introduced so that the variance of total number of bits for all users could be controlled. The cost of fairness is that the better-condition loops have to reduce their data rate, because the worse-condition loops have little improvement in their data rate.

2. Bit loading algorithms for DMT

The application of discrete multi-tone (DMT), divides the frequency selective channel into large number of narrow subchannels, so that each subchannel could be used to modulate a fraction of information in parallel independently. The advantage of DMT is that the power spectral density (PSD) of transmitted signal could be chosen according to the subchannel signal-to-noise ratio (SNR) in order to obtain the optimal performance (e.g. maximum data rate, or minimum power consumption). The problem of how to allocate energy or information (bits) into subchannels optimally is a critical issue in the design of DMT systems. This problem is called bit loading. Many algorithms have been studied in the past for DMT systems such as (Akujuobi, C. M. & Shen, J. 2006), (Yu & Cioffi, 2001), (Hughes-Hartogs, 1987-1989), (Sonalker & Shively, 1998), (Chow & Cioffi, 1995), (Tu & Ciofi, 1990), (Leke & Cioffi, 1997) and (Campello, 1999). They are the foundations for the work discussed in this chapter. A review of several typical bit-loading algorithms are discussed in this Section.

2.1 Channel capacity

In information theory, the capacity of a band-limited white noise channel is: (Cover & Thomas, 1991)

$$C = \frac{1}{2} \log_2 \left(1 + \frac{P}{N_0 W} \right) \text{ bits per sample} \quad (1)$$

where the power of signal is P watts, the spectral density of white noise is $N_0/2$ watts/Hz, and the bandwidth is W Hz. The unit of capacity in Equation (1) is bits per sample, if we convert the unit to bits per second, the expression is the famous channel capacity formula proved by Claude Shannon in 1948 (Shannon, 1948) as shown in Equation (2):

$$C = W \log_2 \left(1 + \frac{P}{N_0 W} \right) \text{ bits per second} \quad (2)$$

In DMT systems, QAM is used as modulation method to map digital information into complex numbers. And we know that QAM is a two-dimensional modulation method, which means it has two basis functions as in-phase function and quadrature function. Therefore, the channel capacity in QAM DMT is

$$C_n = \log_2 (1 + SNR_n) \quad (3)$$

where SNR_n refers to the signal-to-noise ratio for subchannel n , and C_n refers to the capacity of subchannel n . Channel capacity is the theoretic upper limit of achievable data rate for a channel with probability of error that tends to zero. In practical analysis, the probability of error can never be zero; instead, we expect an acceptable error probability P_e at some practical data rate. The reduced data rate could be expressed in a revised channel capacity formula by introducing a SNR gap Γ .

$$b_n = \log_2 \left(1 + \frac{SNR_n}{\Gamma} \right) \quad (4)$$

When $\Gamma = 1$ (0 dB), b_i becomes the channel capacity. The selection of Γ depends on the error probability P_e and coding scheme. Higher P_e requires larger Γ . Complex coding scheme that guarantees reliable transmission can reduce the Γ . For the two-dimensional QAM system with bit error rate (BER) at 10^{-7} , the gap Γ is computed using the following formula: (Chow et al., 1995)

$$\Gamma = 9.8 + \gamma_m - \gamma_c \quad (dB) \quad (5)$$

where γ_m is the performance margin and γ_c is the code gain. If the system is uncoded ($\gamma_c = 0$ dB) and performance margin is 0 dB, the gap is 9.8 dB.

3. Review of bit-loading algorithms

3.1 Water-filling algorithm

Water-filling algorithm is demonstrated in (Cover & Thomas, 1991) and (Gallager, 1968) as the optimal solution for the problem that distributes energy into parallel independent Gaussian channels with a common power constraint. Expand the SNR_n in Equation (4) to the ratio of received signal power $P_n H_n$ and noise power N_n , where P_n is the power allocated to subchannel n and H_n is the channel gain. The number of bits that transmits in a subchannel is expressed as in Equation (6).

$$b_n = \log_2 \left(1 + \frac{P_n H_n}{\Gamma N_n} \right) \quad (6)$$

The problem of bit loading is an optimization problem that allocates power to subchannels. The target of the optimization is to maximize the aggregate number of bits transmitted on all N subchannels under the constraint that the total power should not exceed the power budget P .

$$\begin{aligned} & \text{maximize} \quad \sum_{n=1}^N b_n = \sum_{n=1}^N \log_2 \left(1 + \frac{P_n H_n}{\Gamma N_n} \right) \\ & \text{subject to} \quad \sum_{n=1}^N P_n \leq P \end{aligned} \quad (7)$$

Employ Lagrange multipliers method to solve the optimization problem of Equation (7) with the Lagrangian function $L(P_n)$ as:

$$L(P_n) = \sum_{n=1}^N \log_2 \left(1 + \frac{P_n H_n}{\Gamma N_n} \right) + \lambda \left(\sum_{n=1}^N P_n - P \right) \tag{8}$$

The λ in Equation (8) is called Lagrangian Multiplier. Take the derivative on $L(P_n)$ over the variable P_n and make it equal to 0,

$$\frac{\partial L}{\partial P_n} = \frac{1}{\ln 2 \left(1 + \frac{P_n H_n}{\Gamma N_n} \right)} \cdot \frac{H_n}{\Gamma N_n} + \lambda = 0 \tag{9}$$

Solve this equation to get the power allocation P_n as:

$$P_n = -\frac{1}{\lambda \ln 2} - \Gamma \frac{N_n}{H_n} = C_\lambda - \frac{\Gamma}{g_n} \tag{10}$$

(where $C_\lambda = -\frac{1}{\lambda \ln 2}$, $g_n = \frac{H_n}{N_n}$)

This equation could be rearranged to the following form:

$$P_n + \frac{\Gamma}{g_n} = \text{constant } C_\lambda \tag{11}$$

The variable g_n in Equation (11) expresses the signal-to-noise ratio when unit power is transmitted on subchannel n . It is a unified version of SNR, which is a measurement to indicate the quality of subchannels. We can see from Equation (11) that P_n has the form of “water-filling” distribution. That is, the summation of power transmitted in subchannel n and inverse unified signal-to-noise power ratio (multiplied by Γ) must equal to a constant (C_λ). The $\frac{\Gamma}{g_n}$ can be understood as the terrain of a bowl, power is poured into the bowl like water. The value of C_λ represents the resulting water level. Fig. 1 shows the idea of water-filling.

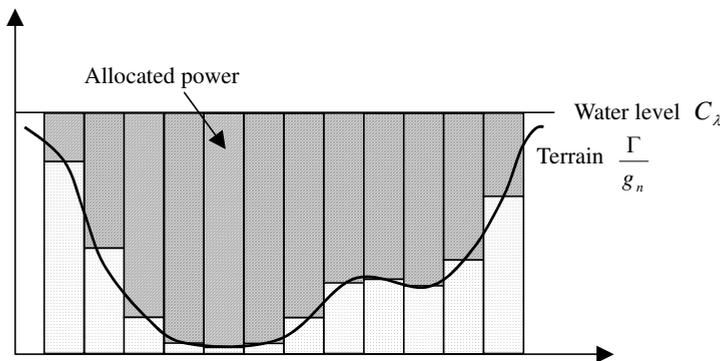


Fig. 1. Water-Filling

The formula to obtain the value of C_λ comes from the constraint of total power budget.

$$P = \sum_{n=1}^N P_n = \sum_{n=1}^N \left(C_\lambda - \frac{\Gamma}{g_n} \right) = NC_\lambda - \Gamma \sum_{n=1}^N \frac{1}{g_n} \quad (12)$$

So we get

$$C_\lambda = \frac{1}{N} \left(P + \Gamma \sum_{n=1}^N \frac{1}{g_n} \right) \quad (13)$$

The water-filling algorithm described in (Starr et al., 1999) and (Tu & Cioffi, 1990) starts from calculating the water-level C_λ , then calculate the power allocation by using,

$$P_n = C_\lambda - \frac{\Gamma}{g_n} \quad (14)$$

The subchannels are sorted in descendent order with respect to the value of g_n . Therefore, if in any step the calculated P_n is negative, which means the subchannel's terrain is too high (signal-to-noise ratio is too small) to hold the power, the algorithm stops. The problem of water-filling algorithm is that in DMT systems, the bit loading for subchannels should be discrete numbers instead of arbitrary real numbers, which is assumed in this algorithm.

3.2 On/Off algorithm – chow's algorithm

The algorithm described in (Chow & Cioffi, 1995) utilizes the fact that if the same or nearly same subchannels are used, the difference of bit-loading result is very small (less than 2%, (Leke & Cioffi, 1997) between the proposed "on/off" algorithm and the traditional water-filling algorithm. In Chow's on/off algorithm, the power distribution is flat, that is, the power is same over all the selected subchannels. On the subchannels that are not selected, the power is simply set to zero, which means they are turned off.

The algorithm starts by sorting the SNRs in descendent order, so that the first subchannel been processed has the highest SNR. At the beginning, the number of subchannels turned on is zero. During each step, one more subchannel is turned on, which causes the total number of turned on subchannels to be K . The power allocated to each turned on subchannel is set to be

$$P_n = \frac{P_{budget}}{K}, \quad (n = 1 \cdots K) \quad (15)$$

where P_{budget} is power constraint. If the P_n is greater than the power mask at subchannel n , the power mask is used as P_n . With the power allocation from Equation (15), numbers of bits in subchannels 1 to n are then calculated using Equation (6). If the aggregated number of bits over all used subchannels becomes less than the value in previous step, the algorithm stops, and the bit allocation scheme obtained from previous step is used as the bit loading result. All the remaining subchannels are thus left in the off state.

The flat power allocation in Chow’s algorithm satisfies the requirement of static spectrum management of ADSL power spectral density. The reason why flat power allocation causes very small difference from optimal water-filling algorithm is studied in detail in (Yu & Cioffi, 2001). The answer is because the logarithm operation in (6) is insensitive to the power actually allocated to subchannels, unless the SNR is small. If the subchannels with SNR less than a cut-off value are turned off, the power could be simply allocated to all other subchannels evenly without loss of much accuracy. An even simpler algorithm was proposed in (Yu & Cioffi, 2001), that save the complexity to find the cut-off point of SNR. When the cut-off SNR is found, power allocation is just to assign constant power to subchannels that has SNRs greater than cut-off value, and assign zero power to other subchannels.

3.3 Greedy algorithm – hughes-hartogs algorithm

In Hughes-Hartogs’s patent (Hughes-Hartogs, 1987-1989), an algorithm based on greedy idea was proposed. That is, every incremental power to transmit one additional bit is allocated to the subchannel that can use it most economically. For example, assume considering only two subchannels *A* and *B*. Subchannel *A* bears N_A bits now, and the incremental power required to transmit N_A+1 bits is ΔP_A . For subchannel *B* that bears N_B bits, the incremental power is ΔP_B . If $\Delta P_A < \Delta P_B$, subchannel *A* will be selected to transmit the additional bit and gets the incremental power allocation. The power requirement for all subchannels to transmit all possible number of bits could be calculated in advance, and be saved in a matrix *P* as show in Fig. 2. The element $P_{m,n}$ in the matrix represents the power needed to transmit *m* bits in subchannel *n*. The values in first row are zeros obviously.

The incremental power required to transmit one additional bit is calculated by subtracting the first row from the second row. The result ΔP is same as the second row in the first iteration. The subchannel *n* that has minimum value in ΔP is selected to get the additional bit. In the next iteration, the elements of column *n* in matrix *P* are shifted upward for one position. The following subtractions are still performed on the row one and row two, until the power budget is fully consumed.

Subchannels	1	2	...	N
0 bit	0	0	...	0
1 bit	$P_{1,1}$	$P_{1,2}$		$P_{1,N}$
2 bits	$P_{2,1}$	$P_{2,2}$		$P_{2,N}$
...	...			
M_{max} bits	$P_{M,1}$	$P_{M,2}$		$P_{M,N}$

Fig. 2. Power Matrix for Hughes-Hartogs Algorithm

The Hughes-Hartogs algorithm was first invented for voice-band modem in 1987, ((Hughes-Hartogs, 1987-1989). For the modern DSL systems, the number of subchannels is usually much larger than the voice-band modems. The slow convergence rate of Hughes-Hartogs algorithm and some other constraints, such as the fixed SNR assumption, make this algorithm impractical in DSL systems. But it is still a good starting point for later improved algorithms.

3.4 Bit removal greedy algorithm

Compared with the bit filling greedy algorithm described in Section 3.3, a bit removal greedy algorithm was proposed in (Sonalker & Shively, 1998). Reversing Equation (6) we can get the power that is required to transmit b_n bits in subchannel n as,

$$P_n[b_n] = (2^{b_n} - 1) \frac{\Gamma N_n}{H_n} = (2^{b_n} - 1) \alpha_n \quad (16)$$

Therefore, the amount of power saved if one bit is removed from the subchannel is

$$\Delta P_n^R = P_n[b_n] - P_n[b_n - 1] = \left[(2^{b_n} - 1) - (2^{b_n - 1} - 1) \right] \alpha_n = 2^{b_n - 1} \alpha_n \quad (17)$$

The bit removal algorithm first allocates the maximum possible numbers of bits to all subchannels. The maximum number of bits in a subchannel is determined by either the power mask limit or the upper limit of allowable bit number – whichever is smaller. Most likely, this bit-loading scheme will exceed the power budget, and the total bits number will be greater than the desired target. Then, the bits are removed one bit per time from the subchannel that may save the power most significantly by removing it. The removing process stops until the power constraint is satisfied, or the data rate requirement is satisfied. Authors (Sonalker & Shively, 1998) made a computation load comparison between bit-removal and bit-filling greedy algorithms over several loops. It showed that in most cases, the bit-removal algorithm required much fewer computations.

3.5 Peter, chow, cioffi, and bingham's practical algorithm

Chow, Cioffi, and Bingham proposed a practical DMT loading algorithm based on the rounded bit number and performance margin adjustment in (Chow et al., 1995). The idea is to make a round operation on the resulting bit loading value, which is expressed as

$$b(n) = \log_2 \left(1 + \frac{SNR(n)}{\Gamma + \gamma_{\text{margin}} (dB)} \right) \quad (18)$$

where γ_{margin} represents the performance margin, and has the initial value of 0. The zero value of γ_{margin} causes the $b(n)$ to get the maximum value. $b(n)$ is then rounded to the nearest integer value $\hat{b}(n)$. In regular condition, the summation of rounded values $\hat{b}(n)$, n from 1 to N , will exceed the target total bit number. In next step, the algorithm increases the margin by using the formula:

$$\gamma_{\text{margin}} = \gamma_{\text{margin}} + 10 \log_{10} \left(2^{\frac{B_{\text{total}} - B_{\text{target}}}{2^{N_{\text{used}} \text{Carriers}}}} \right) \quad (19)$$

With the updated γ_{margin} , Equation (18) is calculated and rounded again. The process continues until the total number of bits reaches the target. If the process doesn't converge after the maximum number of iterations, the algorithm forces the convergence by adjusting the bit allocation according to the difference between $b(n)$ and $\hat{b}(n)$.

At last, the algorithm makes further adjustment on energy distribution so that the bit error rates (BER) on all used subchannels are equal. The analysis in (Chow et al., 1995) shows that only 10 iterations is enough for ADSL loops, which is much faster than the Hughes-Hartogs algorithm.

4. Efficient greedy bit loading with fairness control for multi-user DMT

The twisted-pairs inside a cable could be imagined as a multi-user communication channel because of crosstalk coupling. The received signal power of any user depends on the peering transmitted power, and at the same time is impaired by additive white Gaussian noise (AWGN) and the crosstalk from all other transmitters in the same cable. Fig. 3 shows the multi-user channel environment, where H_{ii} represents the main channel transfer function; H_{ij} ($i \neq j$) represents the crosstalk transfer function from user i to user j ; σ_i represents the power of AWGN noise. Under DMT scenario, we need to study a step further besides the total power of each user. The power allocation over frequency spectrum of each user is of interest.

The power allocation problem relates with the bit-loading problem closely. Actually, they are the two aspects of a same problem, because the number of bits that can be transmitted on a subchannel is a function of the power allocated to that subchannel. Equation (6) gives the relationship between them. In Section 3, we gave a review of several major bit-loading algorithms. However, all those algorithms we discussed are applied to a single modem (or single user). In crosstalk environment, bit-loading algorithms need to be extended to consider mutual effects between multiple users so that the global optimal performance among all modems (users) in a cable could be obtained.

In Section 4, we explore the current available multi-user bit-loading algorithms, and then propose an improved efficient algorithm, which allocates bits to multiple users in parallel on subchannels that have same frequency. This new algorithm reduces the number of iterations dramatically. A new fairness coefficient is also introduced to improve the fairness of data rate among users.

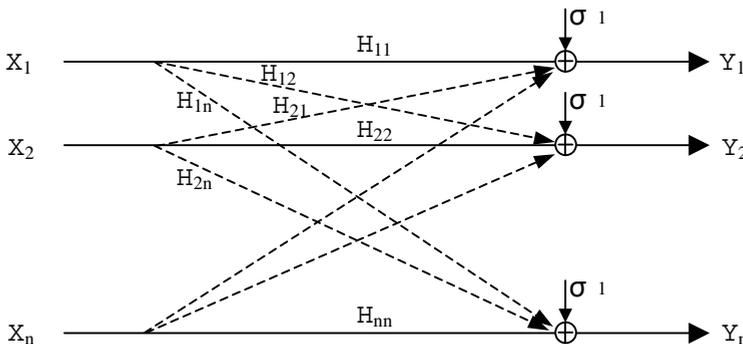


Fig. 3. Multi-user Channel With Crosstalk among Loops

4.1 Notation

The notations that are used in the later Sections of this chapter are defined here. As shown in Figure 4, there are M users in the same cable in which interference exists between users.

We use variables i or j ($i, j = 1, \dots, M$) to indicate the indices of users. In DMT based systems, each user has N sub-channels. The index of sub-channel is indicated by variable n ($n = 1, \dots, N$). We use the term “subchannel (i, n)” to refer to the subchannel n of user i , where $i = 1 \dots M; n = 1 \dots N$. More variables are listed as below:

$b_i(n)$: Number of bits allocated to subchannel (i, n).

$\mathbf{b}(n) = [b_1(n), b_2(n) \dots b_M(n)]^T$: The vector of bit allocations at subchannel n for M users.

$P_i(n)$: Power allocated to subchannel (i, n).

$\mathbf{P}(n) = [P_1(n), P_2(n) \dots P_M(n)]^T$: The vector of power allocation at subchannel n for M users.

$H_{ij}(n)$: Channel gain transfer function from user i to user j at the subchannel n . When $i = j$, $H_{ii}(n)$ is the insertion loss transfer function for user i at subchannel n (ANSI Std. T1.417, 2001). When $i \neq j$, $H_{ij}(n)$ is the crosstalk transfer function from user i to user j at the subchannel n .

Γ : SNR gap margin to ensure the performance under unexpected noise. It is a function of target BER, modulation scheme and coding scheme.

$\sigma_i^2(n)$: The variance of white Gaussian noise on subchannel (i, n).

$P_{mask}(n)$: The power mask on subchannel n .

$P_{budget}(i)$: The power budget for user i .

Δb : The incremental unit of bits added to a subchannel in each iteration. In general, it should be an integer number.

T_{symbol} : The symbol period of the DMT system. For ADSL, it equals to $1/4000$.

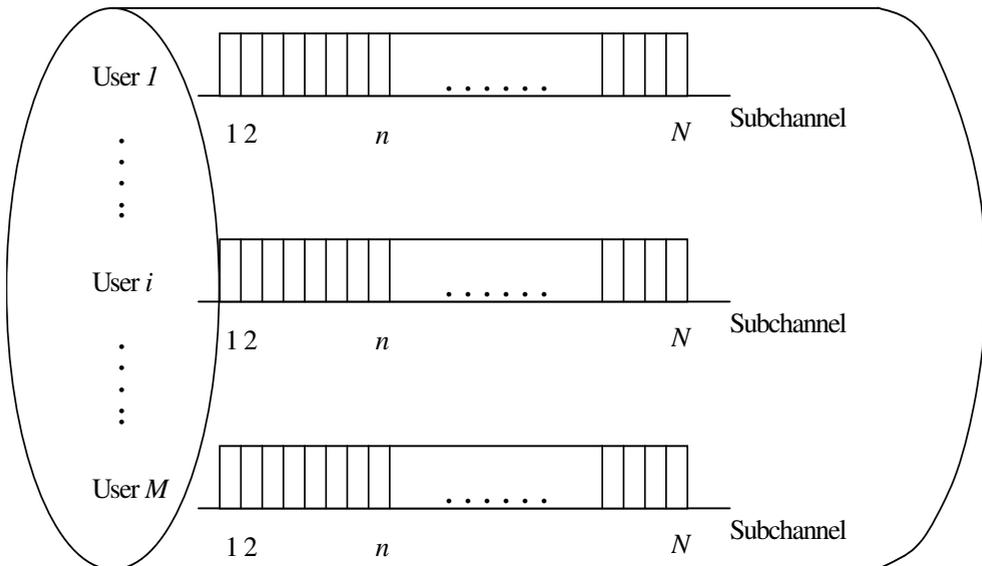


Fig. 4. Multi-User Multi-Channel Configuration

4.2 Problem definition for multi-user bit loading

Like single user bit-loading problem, the problem for multi-user bit loading is an optimization problem. The purpose of the optimization is to find the bit allocation schemes $\mathbf{b}(n)$ for all M users on each subchannel n ($n = 1 \dots N$). The objective is to maximize the aggregate data rate of all users

$$\text{maximize } R_{total} = \sum_{i=1}^M R_i = \frac{1}{T_{symbol}} \sum_{i=1}^M \sum_{n=1}^N b_i(n) \quad (20)$$

with the constraints of power and bit limits, such as:

$$0 \leq P_i(n) \leq P_{mask}(n) \quad (21)$$

$$\sum_{n=1}^N P_i(n) \leq P_{budget}(i) \quad (22)$$

$$b_{min} \leq b_i(n) \leq b_{max}, \text{ and } b_i(n) \text{ is integer} \quad (23)$$

This is called "rate-adaptive loading" (Starr et al., 1999). In some cases, what we care about is not to get the maximum data rate, but to minimize the power consumption with a fixed data rate. The second problem is called "margin-adaptive loading". Formulated as in Equation (24):

$$\begin{aligned} \text{minimize } & \sum_{i=1}^M P_i = \sum_{i=1}^M \sum_{n=1}^N P_i(n) \\ \text{subject to } & b_i = b_{target} \quad (i = 1 \dots M) \end{aligned} \quad (24)$$

These two kinds of problems are equivalent in that algorithms designed for one problem could be applied similarly to another one. In this thesis, we concentrate on the rate-adaptive problem, that is, to maximize the total data rate. As a consequence of maximizing total data rate over all users, the user with better channel condition, which means it has smaller cost to transmit additional bits, will have more chance to get bits assigned to it until it meets the target data rate or exceeds the power budget. The user with worse channel condition is sacrificed in order to gain the maximum total data rate.

However, in most real networks, users in the same cable are of equal priority. They pay the service provider the same fee to get broadband Internet access. Their service quality is supposed to be as equal as possible. Fairness should be considered in the design of bit-loading algorithm. Thus, the multi-user bit loading becomes a multi-objective problem. On one hand, we want maximum total data rate (or equivalently, the minimum power consumption with given target data rate). On the other hand, we want to minimize the difference of data rate among users. Therefore, we defined the second objective as to minimize the variance of data rate among users.

$$\text{minimize } Var(R_i) = \sum_{i=1}^M (R_i - \bar{R})^2 = \sum_{i=1}^M \left(R_i - \frac{1}{M} R_{total} \right)^2 \quad (25)$$

It is clear that the two objectives contradict each other. One direct effect of minimizing the total data rate difference is that the best-condition user cannot obtain its highest data rate as

it can in single user algorithm. They cannot be achieved at the same time. So there must be a tradeoff between them.

4.3 Current multi-user bit-loading algorithms

Distributed iterative water-filling algorithm employs the single-user water-filling method to allocate power iteratively, user by user, until all users and subchannels are filled (Yu et al., 2002). The algorithm runs in two embedded loops. The outer loop looks for the optimal power constraint on each user by increasing or decreasing the power budget for the user, then the inner loop is called to calculate the power allocation under the given power constraint. If the result of inner loop gives data rate lower than target rate, total power will increase, and vice versa. The inner loop employs iterative water-filling method to get optimal power allocation for all the users. The problem of this algorithm is that if the target rate is not appropriate, this algorithm cannot converge. The question then switches to how to obtain the set of achievable target rates for each user. In the coordination of level 1 of DSM, a central agent with knowledge of all channel and interference transfer function exists and is able to calculate the achievable target rates. So, this algorithm is not totally autonomous, some kind of central control is required.

Iterative constant power (ICP) transmission, a slightly variation of iterative water-filling (IW) algorithm is proposed in (Yu & Cioffi, 2001). Both algorithms have the similar two-stage structure. The difference lies in the inner loop: only constant value or zero value of power is allowed in ICP, while continuous power value is used in IW. Both of these two algorithms are suboptimal, but easy to deploy because there is no coordination among users.

The optimal algorithm for discrete multi-user bit loading is a natural extension of single-user greedy algorithm. In the extended greedy algorithm, a matrix of cost is calculated. The elements in the matrix represent the power increment to transmit additional bits for each subchannel and each user. Then, the subchannel in a specific user with minimum cost is found, and additional bits are assigned to it. The process continues until all the power has been allocated. This algorithm is illustrated in (Lee et al., 2002).

A drawback of the multi-user greedy bit loading is the computation complexity. For a single iteration of the algorithm, only one subchannel on one user who has the minimum cost is selected to get additional bits. In each iteration step, the most time consuming calculation is to solve the linear equations to get power allocated to subchannels with specified bits allocation in order to updated the cost matrix. The number of subchannels in a DSL system is usually large, for example, in ADSL there are 223 subchannels for downstream (ANSI Std. T1.417, 2001). If the average number of bits assigned to a subchannel is 10, and there are 50 users in a cable, the total number of iterations that is required to allocate all bits is above 10^5 .

4.4 Formulation of the problem

Before introduce the efficient greedy bit loading with fairness, we first formulate the bit-loading problem for multi-user DMT systems with the objectives of maximizing aggregate data rate (Equation (20)) and minimizing the data rate variance among users (Equation (25)). By extending the single user bit loading to multi-user case, the noise that appears at the

receiver is the summation of AWGN and crosstalk from all other users in the same cable (Fig. 3),

$$N_i(n) = \sigma_i^2(n) + \sum_{\substack{j=1 \\ j \neq i}}^M P_j(n) |H_{ji}(n)|^2 \quad (26)$$

Substitute Equation (26) into Equation (6) to replace the variable N_n , we get the multi-user bit loading expression as

$$b_i(n) = \log_2 \left(1 + \frac{P_i(n) |H_{ii}(n)|^2}{\Gamma \left(\sigma_i^2(n) + \sum_{\substack{j=1 \\ j \neq i}}^M P_j(n) |H_{ji}(n)|^2 \right)} \right) \quad (27)$$

The aggregate data rate of user i is the summation of bits transmitted over all subchannels divided by symbol period,

$$R_i = \frac{1}{T_{symbol}} \sum_{n=1}^N b_i(n) \quad (28)$$

T_{symbol} in Equation (28) is constant, so maximizing Equation (20) is equivalent to maximizing the aggregate bit number over all sub-channels and over all users.

$$b_{total} = \sum_{i=1}^M \sum_{n=1}^N b_i(n) \quad (29)$$

Let us start from the first maximization objective in Equation (29). Constraints (21) and (23) can be used as checking criteria during each step of the optimization. Substitute Equation (27) into Equation (29), we get the first objective function as

$$F_1[\mathbf{P}(n)] = \sum_{i=1}^M \sum_{n=1}^N \log_2 \left(1 + \frac{P_i(n) |H_{ii}(n)|^2}{\Gamma \left(\sigma_i^2(n) + \sum_{\substack{j=1 \\ j \neq i}}^M P_j(n) |H_{ji}(n)|^2 \right)} \right) \quad (30)$$

The constraint is Equation (22). Use the Lagrange multipliers method to solve this problem. Construct the Lagrangian function as

$$L = F_1[\mathbf{P}(n)] + \sum_{i=1}^M \lambda_i \left(\sum_{n=1}^N P_i(n) - P_{budget}(i) \right) \quad (31)$$

Make derivatives of L to $P_i(n)$, and let it equal to zero to find the optimal point of L .

$$\begin{aligned}
 \frac{\partial L}{\partial P_i(n)} &= \frac{1}{\ln 2} \cdot \frac{|H_{ii}(n)|^2 / \Gamma \left(\sigma_i^2 + \sum_{\substack{j=1 \\ j \neq i}}^M P_j(n) |H_{ji}(n)|^2 \right)}{1 + P_i(n) |H_{ii}(n)|^2 / \Gamma \left(\sigma_i^2 + \sum_{\substack{j=1 \\ j \neq i}}^M P_j(n) |H_{ji}(n)|^2 \right)} \\
 &+ \frac{1}{\ln 2} \sum_{\substack{j=1 \\ j \neq i}}^M \frac{-P_j(n) |H_{ii}(n)|^2 |H_{ji}(n)|^2 / \Gamma \left(\sigma_j^2 + \sum_{\substack{m=1 \\ m \neq j}}^M P_m(n) |H_{jm}(n)|^2 \right)^2}{1 + P_j(n) |H_{jj}(n)|^2 / \Gamma \left(\sigma_j^2 + \sum_{\substack{m=1 \\ m \neq j}}^M P_m(n) |H_{jm}(n)|^2 \right)} \\
 &+ \lambda_i \\
 &= 0
 \end{aligned} \tag{32}$$

The first term is the contribution from user i itself. The second term is a function of $P_j(n)$ of all other $(M-1)$ users, and shows the effect of crosstalk. Since this term has a high order in denominator and because the crosstalk is weak compare to the main signal, we can ignore the second term to make the equation tractable and get the approximated expression of $P_i(n)$.

$$\frac{1}{\ln 2} \cdot \frac{|H_{ii}(n)|^2 / \Gamma \left(\sigma_i^2 + \sum_{\substack{j=1 \\ j \neq i}}^M P_j(n) |H_{ji}(n)|^2 \right)}{1 + P_i(n) |H_{ii}(n)|^2 / \Gamma \left(\sigma_i^2 + \sum_{\substack{j=1 \\ j \neq i}}^M P_j(n) |H_{ji}(n)|^2 \right)} + \lambda_i = 0 \tag{33}$$

Making a simple arrangement, $P_i(n)$ is expressed as,

$$P_i(n) = \frac{-1}{\lambda_i \ln 2} - \frac{\Gamma}{|H_{ii}(n)|^2} \left(\sigma_i^2 + \sum_{\substack{j=1 \\ j \neq i}}^M P_j(n) |H_{ji}(n)|^2 \right) \tag{34}$$

We denote

$$C_{i,\lambda} = \frac{-1}{\lambda_i \ln 2} \tag{35}$$

and

$$T_i(n) = \frac{\Gamma}{|H_{ii}(n)|^2} \left(\sigma_i^2 + \sum_{\substack{j=1 \\ j \neq i}}^M P_j(n) |H_{ji}(n)|^2 \right) \tag{36}$$

then $P_i(n)$ is expressed as

$$P_i(n) = C_{i,\lambda} - T_i(n) \tag{37}$$

Equation (37) has the form of “water filling” solution with the water level as $C_{i,\lambda}$, and the terrain that holds the poured power as $T_i(n)$. The constant $C_{i,\lambda}$ can be solved from the constraint (22). Substitute Equation (37) into Equation (22),

$$\sum_{n=1}^N \left[C_{i,\lambda} - \frac{\Gamma}{|H_{ii}(n)|^2} \left(\sigma_i^2 + \sum_{\substack{j=1 \\ j \neq i}}^M P_j(n) |H_{ji}(n)|^2 \right) \right] = P_{budget}(i) \quad (38)$$

$$NC_{i,\lambda} - \sum_{n=1}^N \frac{\Gamma}{|H_{ii}(n)|^2} \left(\sigma_i^2 + \sum_{\substack{j=1 \\ j \neq i}}^M P_j(n) |H_{ji}(n)|^2 \right) = P_{budget}(i) \quad (39)$$

Therefore, we get

$$C_{i,\lambda} = \frac{1}{N} \left[P_{budget}(i) + \sum_{k=1}^N \frac{\Gamma}{|H_{ii}(k)|^2} \left(\sigma_i^2 + \sum_{\substack{j=1 \\ j \neq i}}^M P_j(k) |H_{ji}(k)|^2 \right) \right] \quad (40)$$

According to (Marler & Arora, 2003), the solution of a multi-objective optimization could be obtained by the “Lexicographic Method”. The method solves the objective functions in the order of importance, and constructs new constraints for next objective function using the solution for previous objective. In this multi-user bit-loading problem, we first get the optimal solution $\mathbf{b}^*(n)$ of Equation (29) as described above, and then process the second objective by minimizing Equation (25) and subject to

$$b_{total}[\mathbf{b}(n)] \leq b_{total}[\mathbf{b}^*(n)] \quad (41)$$

It is obvious that this optimization is even more complex than the first objective optimization, because both objective function and constraint function include variables in high order terms and denominators. A practical implementation of this algorithm is required.

4.5 Greedy algorithm for multi-user nit loading

A practical method to obtain the optimal solution of Equation (29) is the extended greedy algorithm. The foundation of greedy algorithm is to calculate the power cost to transmit additional bits for each subchannel (i, n). First, we rearrange Equation (29) to remove the logarithm,

$$\Gamma(2^{b_i(k)} - 1) = \frac{P_i(k) |H_{ii}(k)|^2}{\sigma_i^2 + \sum_{\substack{j=1 \\ j \neq i}}^M P_j(k) |H_{ji}(k)|^2} \quad (42)$$

Let,

$$f(b_i(n)) = \Gamma(2^{b_i(n)} - 1) \quad (43)$$

Then we get,

$$P_i(n) - f(b_i(n)) \sum_{j=1, j \neq i}^M P_j(n) \frac{|H_{ji}(n)|^2}{|H_{ii}(n)|^2} = f(b_i(n)) \frac{\sigma_i^2}{|H_{ii}(n)|^2} \quad (44)$$

The above equation can be expressed in matrix form as

$$AX = B \quad (45)$$

where

$$A = \begin{bmatrix} 1 & \dots & \frac{-f(b_1(k))|H_{i1}(k)|^2}{|H_{11}(k)|^2} & \dots & \frac{-f(b_1(k))|H_{M1}(k)|^2}{|H_{11}(k)|^2} \\ \vdots & 1 & \dots & \dots & \vdots \\ \frac{-f(b_i(k))|H_{ii}(k)|^2}{|H_{ii}(k)|^2} & \dots & 1 & \dots & \frac{-f(b_i(k))|H_{Mi}(k)|^2}{|H_{ii}(k)|^2} \\ \vdots & \dots & \dots & 1 & \vdots \\ \frac{-f(b_M(k))|H_{iM}(k)|^2}{|H_{MM}(k)|^2} & \dots & \frac{-f(b_M(k))|H_{iM}(k)|^2}{|H_{MM}(k)|^2} & \dots & 1 \end{bmatrix} \quad (46)$$

$$X = [P_1(k) \quad \dots \quad P_i(k) \quad \dots \quad P_M(k)]^T \quad (47)$$

$$B = \begin{bmatrix} \frac{f(b_1(k))\sigma_1^2}{|H_{11}(k)|^2} & \dots & \frac{f(b_i(k))\sigma_i^2}{|H_{ii}(k)|^2} & \dots & \frac{f(b_M(k))\sigma_M^2}{|H_{MM}(k)|^2} \end{bmatrix}^T \quad (48)$$

In Solving the above linear equation system, we obtain the power vector $\mathbf{P}(n)$ required to transmit bit allocation scheme $\mathbf{b}(n)$ on subchannel n for all M users. One thing worthy of notice is that the solution of $\mathbf{P}(n)$ vector may contain negative elements. This indicates that the corresponding users cannot afford to transmit the number of bits assigned to them on subchannel n .

In DMT, different subchannels are well separated. The crosstalk coupling between different users only appears in the subchannels with same frequency. So we assume there is no interference between subchannels. Equation (45) with different $n = 1, \dots, N$ could be solved independently. Let $b'_i(k) = b_i(k) + \Delta b$ for $i = 1$ to M and $n = 1$ to N , do a calculation of Equation (45) again to obtain the new power allocation $\mathbf{P}'(n)$. Because of the crosstalk coupling, one element change in the coefficient matrix A in Equation (45) causes power of all the users on the same subchannel to change. The cost of adding Δb bits on subchannel (i, n) is the summation of power increment on all users on subchannel n .

$$\text{cost}(i, k) = \sum_{i=1}^M (P'_i(k) - P_i(k)) \quad (49)$$

This calculation needs to be done on every subchannel and every user. The final cost matrix is M rows by N columns, with each element at position (i, n) represents the cost of transmitting additional bits on subchannel n of user i . The position of the minimum cost determines the subchannel and user where additional bits will be transmitted.

Power budget need to be checked during the cost updating. Additional Δb bits added to subchannel n of user i may cause the user i to get more power on subchannel n , and at the same time, it causes all other users to increase their power on subchannel n in order to maintain their SNR to transmit already-assigned number of bits. Either the user i or other users may have the possibility to exceed their power budget. If this happens, it means that the adding of Δb bits to subchannel n of user i is not feasible.

4.6 Efficiency improvement to add bits to multiple users in parallel

As discussed in the Section 4, the large number of iterations for multi-user DMT makes the greedy bit-loading algorithm hard to deploy. We mitigate this problem by processing multiple users in parallel, so that the number of iterations could be reduced dramatically. As we know, crosstalk only interferes with users on the subchannels that are in same frequency. Different-frequency subchannels are independent with each other. In other words, we can rewrite Equation (45) by indicating the subchannel index n explicitly as

$$A(n)X(n) = B(n) \quad (50)$$

If, for example, two subchannels with different indices n_1 and n_2 get additional bits, no matter whether the subchannels are for the same user or for different users, two linear Equations of (50) have to be calculated. This means that adding bits to subchannels in the dimension of subchannel index requires the same number of iterations as the number of subchannels processed. However, if we add bits to subchannels in the dimension of user index with subchannel index fixed, it is possible to reduce the number of calculations of Equation (50). Let us say, for instance, within a specific subchannel n , we assign additional bits to two users i_1 and i_2 , the resulting power scheme at subchannel n $\mathbf{P}'(n)$ could be calculated by solving a single equation of Equation (50).

The subchannel with minimum cost is identified by both subchannel index n and user index i . So, in the proposed algorithm, instead of adding bits to only one subchannel (i, n) , we look for all users on subchannel n , which has cost very close to the minimum cost (cost_{\min}). The additional bits are added to subchannel n of all these users. Fig. (5) visualizes the idea. The term "close" to the minimum cost is defined by a cost elastic coefficient δ_{cost} . On subchannel n where cost_{\min} appears, if any user i satisfies the condition

$$\frac{\text{cost}(i, n) - \text{cost}_{\min}}{\text{cost}_{\min}} < \delta_{\text{cost}} \quad (51)$$

we add additional bits to it. The value of δ_{cost} shows the percentage degree of how much the cost on a given subchannel is greater than the minimum cost. It could have any value greater than zero, depending on the accuracy we want in the algorithm. The effect of this coefficient will be analyzed in next chapter. The simulation result shows that importing this cost elastic coefficient has nearly no negative impact on the final bit-loading scheme, but the number of iterations reduced greatly.

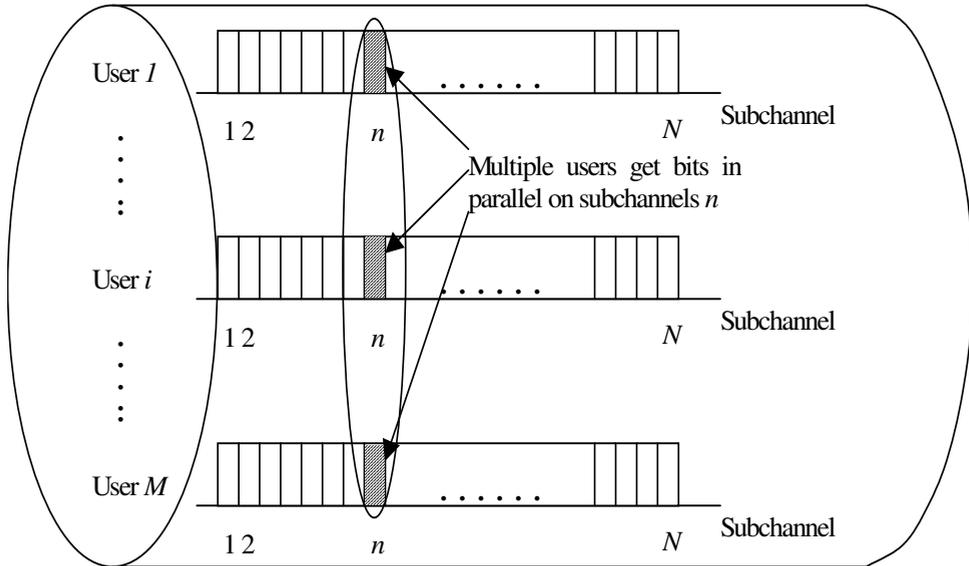


Fig. 5. Multiple Users Get Bits in Parallel

With the updated bit-loading scheme on subchannel n , we solve a single linear equation set in Equation (50) to obtain the new power allocation scheme. It reflects the changes of bit number for several users. To prepare for the next iteration, the n th column in cost matrix needs to be updated. As usual, we assume additional bits are added to each user at subchannel n , and calculate the cost according to Equation (49). The comparison of traditional algorithm and improved efficient algorithm is listed in Table 1.

Traditional Algorithm	Efficient Algorithm
<p>Do</p> <p>Update cost matrix</p> <p>Find cost_{\min} at subchannel (i, n)</p> <p>Add bit to subchannel (i, n)</p> <p>Update nth column of A, B, X in (50)</p> <p>While all subchannels are filled or power budget are reached</p>	<p>Do</p> <p>Update cost matrix</p> <p>Find cost_{\min} at subchannel (i_0, n)</p> <p>Find all users (i_1, \dots, i_k) that have cost close to cost_{\min} at subchannel n</p> <p>Add bit to users i_0, i_1, \dots, i_k at subchannel n</p> <p>Update nth column of A, B, X in (50)</p> <p>While all subchannels are filled or power budget are reached</p>

Table 1. Algorithm Comparison of Traditional and Efficient Multi-User Bit Loading

4.7 Fairness control

The objective of minimizing the data rate variance among M users in Equation (25) is equivalent to minimize the variance of total number of bits of M users. We obtain this objective by importing a fairness elastic coefficient δ_{fair} in the bit loading process. In each greedy bit loading iteration, we check the total bits number $\text{sum_bits}(i)$ of the user that just

got the additional bits assigned to it, and keep the number in lock with the averaged total bit numbers of all the other users. The range of lock is adjustable by using the fairness control coefficient δ_{fair} .

$$\text{sum_bits}(i) \leq \delta_{\text{fair}} \frac{1}{M-1} \sum_{\substack{j=1, j \neq i \\ j \notin \text{Frozen}}}^M \text{sum_bits}(j) \quad (52)$$

When the $\text{sum_bits}(i)$ exceeds the average sum_bits of all other users, the user i is to be in “frozen” state temporarily. Users in frozen set will not participate in the further bits assignment until they are unfrozen. The value of fairness elastic coefficient δ_{fair} depends on the degree of fairness we want. $\delta_{\text{fair}} = 1$ ensures the final sum_bits of all users have minimum variance. If we loose the elastic coefficient, the variance may increase, and the better-condition users can get higher data rate.

4.8 Algorithm Implementation

In this work, we designed our newly proposed algorithm as a flexible solution, so that it supports both parallel bit loading and fairness adjustment, or other kinds of combinations. For example, we could run the algorithm with only parallel bit loading, or only fairness adjustment, or neither of them, which becomes the traditional multi-user greedy bit loading. This flexibility makes it easier for us to run the algorithm in different modes, and observe the effects of our improvements.

Besides the switch variables that control the mode of algorithm, two flag variables (flag_user and flag_channel) are used to assist in the algorithm execution. The variable flag_user is a vector that indicates the status of M users. The user i is available in bit allocation process if $\text{flag_user}(i) = 0$; If the user i is set to “frozen” status because of the fairness adjustment, $\text{flag_user}(i)$ has the value of 1. The user will relinquish its chance to get additional bits on all its subchannels, until the $\text{flag_user}(i)$ is set to 0 again. Since multi-user bit loading involves two dimensions - users and subchannels, we defined a matrix flag_channel (M rows and N columns) to indicate whether a subchannel (i, n) is available to allocate more bits. $\text{flag_channel}(i, n) = 0$ indicates subchannel n of user i is available; $\text{flag_channel}(i, n) = 1$ indicates this subchannel is “full”. There are two possible reasons that cause a subchannel to be full:

1. The number of bits assigned to the subchannel reaches the cap b_{max} ;
2. Adding additional bits causes the power consumed on user i to exceed the power budget.

The algorithm is described in the following list.

Initialization

1. $b_i(n) = \mathbf{O}_{M,N}$; ($\mathbf{O}_{M,N}$ is the M rows N columns zero matrix)
2. $P_i(n) = \mathbf{O}_{M,N}$;
3. $\text{flag_user} = \mathbf{O}_{1,M}$;
4. $\text{flag_channel} = \mathbf{O}_{M,N}$;
5. $\text{sum_bits} = \mathbf{O}_{1,M}$;
6. $\text{cost}(i, n) = \mathbf{O}_{M,N}$;
7. Calculate the initial matrices A and B using (46) and (48);

8. Let additional bits to be assigned to each subchannel temporarily and calculate the new power vector $\mathbf{P}'(n)$ using Equation (45), then calculate the initial cost matrix using Equation (49);

Iteration

1. Find the subchannel (i, n) that has the minimum value in cost matrix.
2. If the minimum cost is greater than or equal to a predefined BIGNUMBER, stop the iteration; algorithm finishes because no subchannel can get additional bits without breaking the power budget.
3. If parallel bit loading is allowed, find all users that have costs close to the minimum cost to transmit addition bits on subchannel n .
4. Add Δb bits to subchannel n of all the users selected in step 3.
5. Update the matrices A and B for subchannel n , and calculate the new power vector $\mathbf{P}'(n)$.
6. Check if there are any elements in $\mathbf{P}'(n)$ that have negative value. If so, consider two cases: 1. multiple users got additional bits on subchannel n , then mark this subchannel to not allow parallel bit loading and return to step 1; 2. only one user got additional bits, set $\text{flag_channel}(i, n) = \text{"full"}$, and set the $\text{cost}_i(n)$ to be BIGNUMBER, then return to step 1.
7. Check if the total power on any user exceeds the budget. If so, do the same processing as in step 6.
8. If fairness adjustment is allowed, check the fairness using Equation (52). If the bit sum of user i exceeds the elastic range, set $\text{flag_user} = \text{"frozen"}$, and add a BIGNUMBER to the costs of all subchannels on this user.
9. For those users that already belong to the frozen set, check if their bit sum drop back into the elastic range. If so, unfreeze them, and subtract BIGNUMBER from subchannel costs to restore the original values.
10. Update part of cost matrix, which correspond to subchannel n of all users.
11. Go to step 1.

5. Simulation, results and analysis

We implemented the proposed multi-user bit-loading algorithm for DMT in MATLAB. In Section 5, we analyzed the performance of our algorithm and made comparisons for different variances. The simulation was applied on ADSL and straight loop with variant loop lengths. The models of loop and far-end crosstalk (FEXT) came from (ANSI Std. T1.417, 2001). We assume there are 50 subscriber loops in a cable, so the number of FEXT sources is 49. The loop lengths are generated as uniformly distributed random numbers within the range of 2 - 16 kft. They are sorted in ascendant order so that the performance of these loops would be displayed in descending order. Fig 6 shows the sorted loop lengths that were used in our simulation.

Only downstream signals were considered for simplicity in this simulation. The process of upstream signal is similar. As shown in Fig. 7, each receiver at the far end of the loop gets signal from both its own transmitter and FEXT noise from all other transmitters in the same cable. We assume all the 50 loops in the cable transmit ADSL signals; and no interference from other services exists.

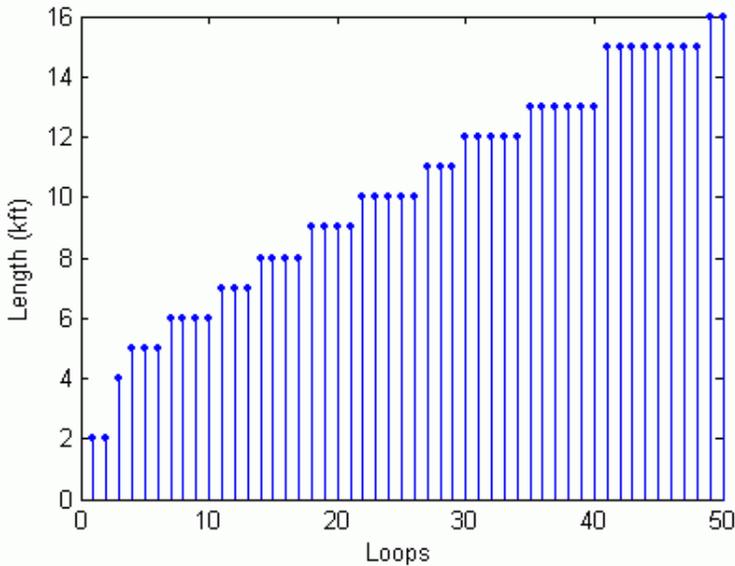


Fig. 6. Loop Lengths for the Simulation

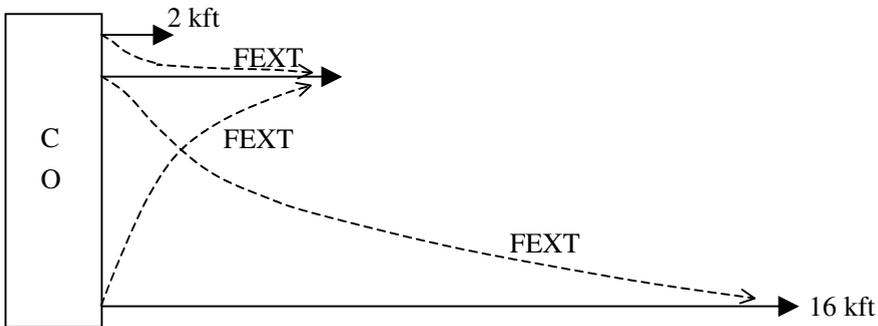


Fig. 7. FEXT Among Loops with Different Lengths

We assume there is no code gain and no noise margin, so the SNR gap Γ is chosen as 9.8 dB according to Equation (5). In the downstream of ADSL DMT, the subchannels start from 33 and end at 255 (ANSI Std. T1.417, 2001). Therefore, we are considering $255-33+1=223$ subchannels in this simulation. The corresponding frequency range is 142 kHz - 1100 kHz with subchannel spacing as 4312.5 Hz. The power budget for each user is 20.4 dBm (ANSI Std. T1.417, 2001). For simplicity, we chose the incremental unit for bit addition as $\Delta b = 1$ bit.

5.1 Simulation mode

We have two improvements in the proposed algorithm: parallel bit allocation and fairness control. So we run the algorithm in all possible combinations. Table 2 shows the four operational modes of the algorithm. In traditional mode, neither of the two new features are applied, this is equivalent to the traditional multi-user DMT bit loading as in (Lee et al., 2002). We used this mode as reference to compare with our proposed features.

	Parallel Bit Allocation	Fairness
Traditional Mode	N	N
Parallel Only Mode	Y	N
Fairness Only Mode	N	Y
Both Mode	Y	Y

Table 2. Operational Mode of the Algorithm

5.2 Simulation result analysis

In the proposed algorithm, we introduced two elastic coefficients (δ_{cost} and δ_{fair}) for parallel bit loading and fairness control respectively. However, how the coefficients affect the performance of algorithm is not straightforward. This section shows the analysis of the simulation results to demonstrate that by using cost elastic coefficient, the computational load decreased significantly; and by using fairness elastic coefficient, we have a way to control the data rate fairness among users in the same cable.

Traditional operational mode is used as comparison reference. In the following analysis, the value of -1 for δ_{cost} or δ_{fair} indicates that the algorithm does not utilize the proposed parallel bit loading or fairness control. The first data points in the following figures that have control value of -1 are used as reference points.

5.3 Effectiveness of parallel bit loading

The purpose of parallel bit loading is to reduce the computational load in the traditional multi-user DMT bit loading. In this section, we ran the algorithm with no fairness control so that it has comparability with traditional operational mode in terms of parallel bit loading. From Fig. 8 we see that when there are 50 users in a cable, the number of iterations required

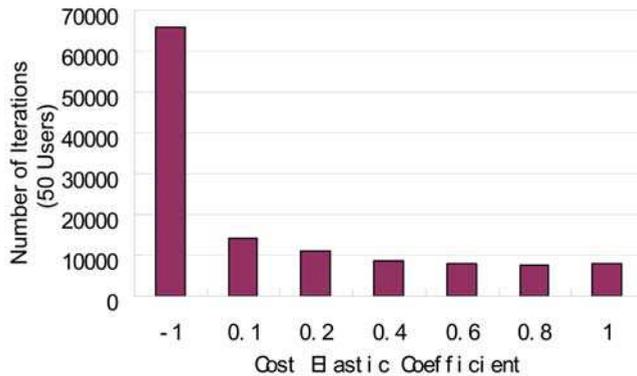


Fig. 8. Number of Iterations vs. Cost Elastic Coefficient (50 Users)

to allocate all bits to users and subchannels is 66016, if no parallel bit-loading technique is used ($\delta_{cost} = -1$). If δ_{cost} is set to 0.1, the iterations number dropped immediately to only 22% of the previous value. The number even reduced to 12% if the δ_{cost} is set to be greater than 0.6. This shows that parallel bit loading has a great improvement on the computational load for multi-user DMT bit loading. Fig. 9. shows the similar effect with the comparison for both 50 users and 15 users. We see that the more users in a cable, the more significant improvement could be achieved for number of iterations. Furthermore, we noticed that as

long as parallel bit loading is used, the iteration numbers is not very sensitive to the value of δ_{cost} , especially when $\delta_{cost} \geq 0.4$. Set δ_{cost} to 1 is a good choice.

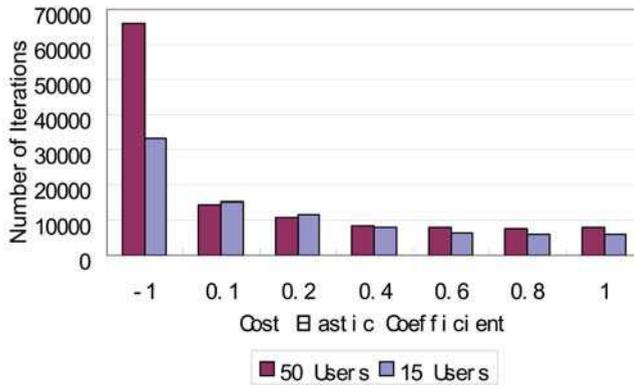


Fig. 9. Number of Iterations vs. Cost Elastic Coefficient (50 Users and 15 Users)

Reducing the iterations number is not the only goal of the algorithm. To make the algorithm meaningful, we must guarantee that the final bit-loading result has no or only little loss comparing with the traditional algorithm. We run the algorithm to generate bit-loading schemes for 50 users. The number of bits on each user over all subchannels determines the data rate of the user. The average value (mean) of the bits numbers for 50 users specifies the performance of bit-loading algorithm.

$$\bar{b} = \frac{1}{50} \sum_{i=1}^{50} \sum_{n=1}^N b_i(n) \tag{53}$$

Table 3 and Fig. 10. show the mean bit numbers when different δ_{cost} values were used. It is clear from the figure that the mean bits numbers are constant for all δ_{cost} values that are less than or equal to 1. As cost elastic coefficient increases to greater than 1, the mean bit number decreases. Therefore, selecting δ_{cost} to be 1 does not result in loss of any accuracy of the algorithm.

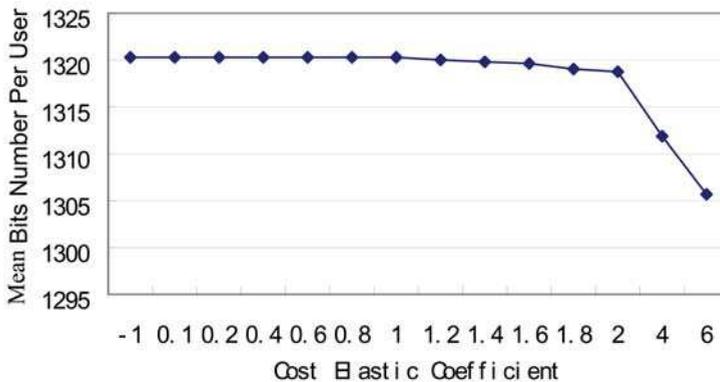


Fig. 10. Mean Bits Number Per User vs. Cost Elastic Coefficient

Fig. 11 demonstrates that many users get bits assigned to them simultaneously in a single iteration (marked by *). In this case, bits were assigned to 32 users in the iteration 3000. In other words, one iteration in the new efficient algorithm is equivalent to 32 iterations in traditional algorithm. That is a remarkable improvement.

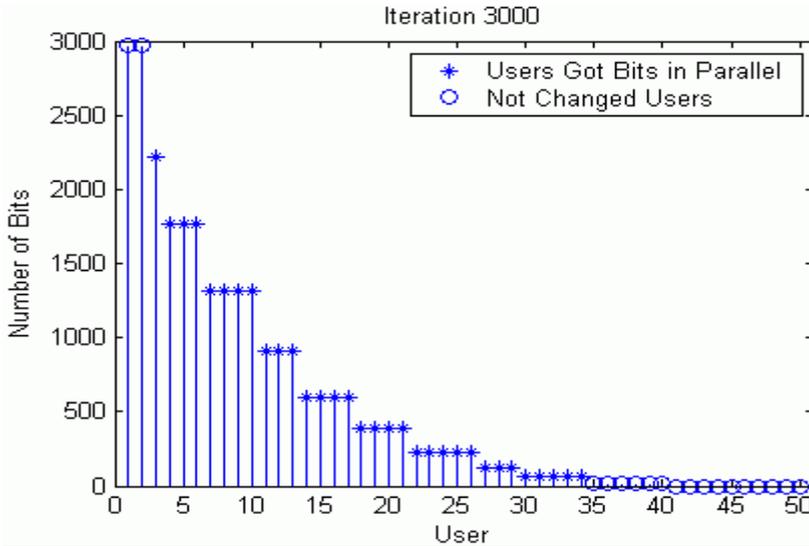


Fig. 11. Multiple Users Get Bits in a Single Iteration

5.4 Effectiveness of fairness control

By introducing the fairness control coefficient δ_{fair} , we have the ability to constraint the variance of bit numbers over all users. As stated before, the loop lengths of 50 users have been sorted in ascendant order. That is, user 1 has the shortest loop length, and user 50 has the longest loop length. Therefore, if no fairness control is applied, the total number of bits assigned to each user is in the descendant order. The dashed curve in Fig. 13 is in this case. When we set the δ_{fair} to be 1, the bits number of each user is limited to have minimum variance. Actually, in Fig. 12, it is a straight line (the solid curve). Two more curves are showed in Fig. 12, which represent the case for δ_{fair} equals to 1.2 and 1.4 respectively. Fig. 13 shows the mean and standard deviation of bits number per user in the same figure when different δ_{fair} were used. We see that these two curves have similar shapes. That is, when the standard deviation reduced, the mean also reduced. The benefit of fairness is compensated by the loss of average data rate. Fig. 14 shows how the fairness control coefficient affects the number of iterations. When δ_{fair} equals to 1, the number of iterations is minimum, but at the same time, the number of bits assigned to each user is also minimum. Fig. 12 also shows that the bits numbers of short-loop users have very limited improvement when fairness control is applied. So the bits numbers for short-loop users have to be dropped greatly to obtain the small variance. This indicates that although the application of fairness control makes the standard deviation of bits number (data rate) small, we pay the cost to sacrifice the performance for short-loop users. Therefore, the fairness control may not be strongly desired.

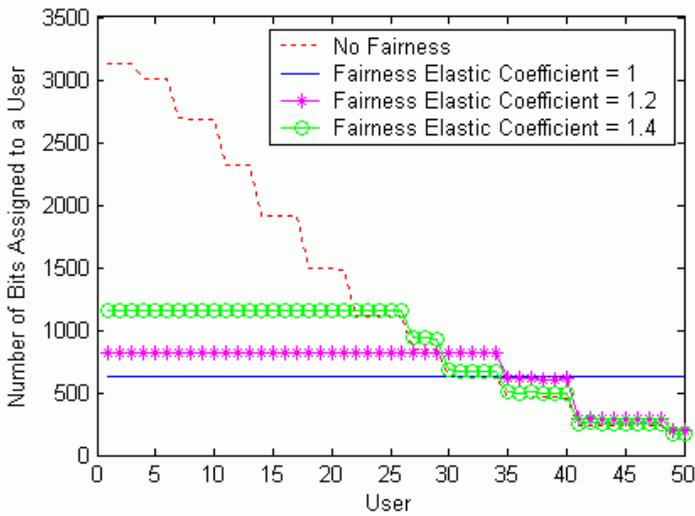


Fig. 12. Bits Number of Users

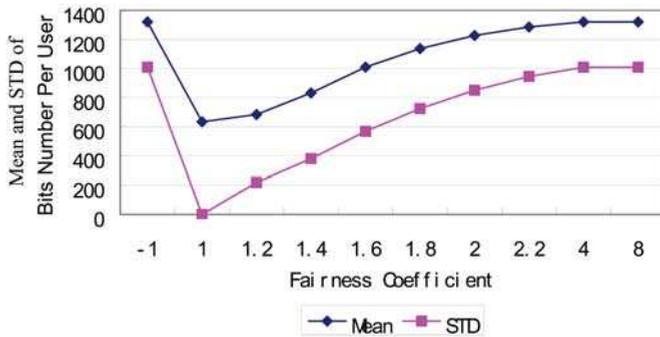


Fig. 13. Mean and Standard Deviation of Bits Number vs. Fairness Coefficient

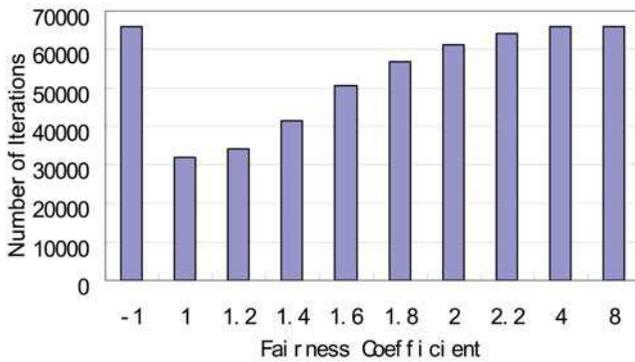


Fig. 14 Number of Iterations vs. Fairness Coefficient

5.5 Combination of parallel bit loading and fairness control

When the parallel bit loading and fairness control are applied at the same time, the combined effectiveness of them needs to be identified. This section analyzed the simulation results and illustrated them in 3-D plots. The algorithm was run in full combination of δ_{cost} and δ_{fair} . The major performance measures are recorded, such as number of iterations, mean and standard deviation of bits number per user. Tables 3, and 4 are some of the simulation results.

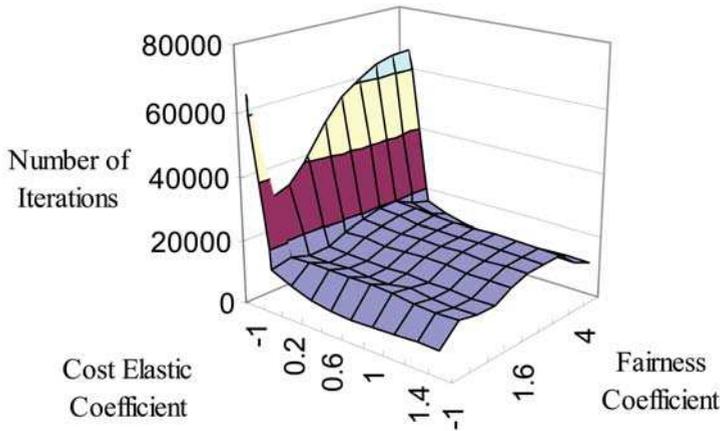


Fig. 15. Number of Iterations vs. Cost Elastic Coefficient and Fairness Coefficient

Fig. 15 indicates that the number of iterations has a strong correlation with cost elastic coefficient. As soon as the parallel bit loading is applied (δ_{cost} has any value other than -1), the number of iterations dropped significantly. Compared with the significance of parallel bit loading, the fairness coefficient has relatively little contribution to reduce the iteration number.

$\delta_{cost} \backslash \delta_{fair}$	-1	0.1	0.2	0.4	0.6	0.8	1
-1	1320.3	1320.3	1320.3	1320.3	1320.3	1320.3	1320.3
1	632.96	632.96	632.96	632.96	632.96	632.96	629.98
1.2	683.4	703.48	703.46	705.1	701.92	696.02	719.48
1.4	831.48	831.52	831.46	830.78	830.1	829.96	829.38
1.6	1009.22	1009.22	1009.24	1008.76	1009.2	1008.82	1009.36
1.8	1136.78	1137.18	1137.16	1137.7	1138.2	1138.26	1138.38
2	1226.76	1227.2	1226.98	1228.22	1228.9	1229.32	1229.4
2.2	1285.82	1286	1286.02	1287.18	1287.8	1287.74	1287.8
4	1320.3	1320.3	1320.3	1320.3	1320.3	1320.2	1320.2
8	1320.3	1320.3	1320.3	1320.3	1320.3	1320.3	1320.3

Table 3. Mean Bits Number Per User Under Different Coefficients Values

$\delta_{\text{cost}} \backslash \delta_{\text{fair}}$	-1	0.1	0.2	0.4	0.6	0.8	1
-1	1008.1713	1008.1713	1008.1713	1008.1713	1008.1713	1008.1713	1008.1713
1	0.19795	0.19795	0.19795	0.19795	0.19795	0.19795	0.14142
1.2	218.5638	223.4621	223.5784	224.1786	223.0507	221.6587	229.2879
1.4	380.3019	380.3123	380.3548	379.924	379.6629	379.6214	379.1612
1.6	567.4984	567.5644	567.6492	567.3555	568.2036	567.9887	568.7456
1.8	723.7036	724.4047	724.506	725.7295	726.8624	727.1101	727.4138
2	850.2531	850.9939	850.6976	852.8352	854.0387	854.7548	854.9612
2.2	944.991	945.3436	945.3933	947.5979	948.9535	948.8161	948.8768
4	1008.1713	1008.1713	1008.1713	1008.1713	1008.1713	1007.935	1007.9803
8	1008.1713	1008.1713	1008.1713	1008.1713	1008.1713	1008.1713	1008.1713

Table 4. Standard Deviation of Bits Number Per User Under Different Coefficients Values

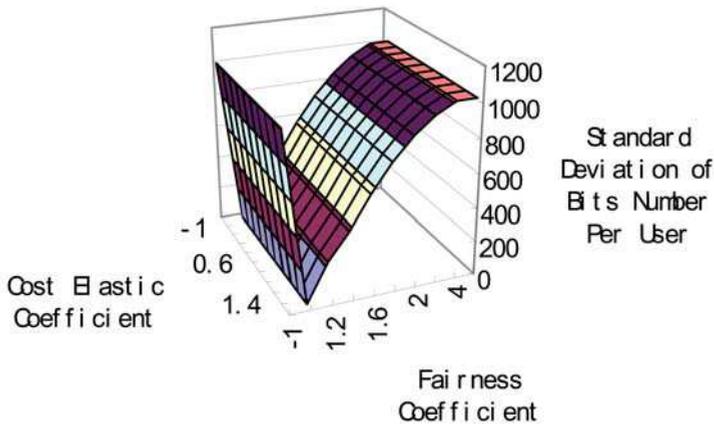


Fig. 16. Standard Deviation of Bits Number vs. Cost Elastic Coefficient and Fairness Coefficient

In the plot of Table 3 which is not shown here, it shows how the mean bits number per user is affected by cost elastic coefficient and fairness coefficient. The surface is flat along the cost elastic coefficient direction, which means the selection of δ_{cost} has almost no influence on the mean bits number. The only contribution to affect of the mean bits number is the value of δ_{fair} . Fig.16 shows how cost elastic coefficient and fairness coefficient affect the standard deviation of bits number per user. The isolated effects of parallel bit loading and fairness control gave us the flexibility to apply them separately or in combination according to the requirement.

6. Summary of conclusions

Traditional bit-loading algorithms for DMT system were designed for a single user. As more and more users turn to DSL services, crosstalk inside the cable makes the traditional single-user bit-loading algorithms unable to reach the optimal solution. Multi-user bit-loading algorithms were designed to obtain the global optimization among all users in a cable. The problem of multi-user bit loading was computational complexity. Because the number of subchannels for all users is very large, the calculation of power allocation needs to be done many times.

This research work studied several bit-loading algorithms for both single user and multiple user cases. Then an improved parallel bit-loading algorithm for multi-user DMT was proposed. The new algorithm was based on multi-user greedy bit loading. In a single iteration, the bits were allocated to multiple users on the subchannels with same frequency. Thus, the number of iterations to allocate bits to all users decreased significantly. The adjustable cost elastic coefficient defined the range of power cost. In the bit-loading iteration, all subchannels for different users that have the power cost within the range have the chance to get additional bits assigned to them. Furthermore, this research work studied the possibility to improve the fairness among users in the same cable. We introduced a fairness coefficient during bit-loading iterations so that the variance of total number of bits for all users could be controlled.

The analysis of simulation results showed that the effectiveness of applying parallel bit loading is significant. If the cost elastic coefficient has the value of 1, it means that all subchannels who has power cost less than twice of the minimum cost could get additional bits assigned to them. For 50 users case and $\delta_{\text{cost}}=1$, the number of iterations to load all bits to all users reduced to only 12% of the number if no parallel bit loading is used. Another good characteristic of parallel bit loading is that it has very small effect on the final loading result. In the same 50-user case, when δ_{cost} is less than 1, the loading result is exactly the same as no parallel bit loading. This means that we reduced the computational complexity without losing any accuracy of the result. The simulations also showed that the fairness in the algorithm control could limit the variance of total bit numbers among all users. The cost of fairness is that the better-condition loops (for example, shorter loop length) have to reduce their data rate, because the worse-condition loops have little improvement in their data rate.

7. Recommendations for future work

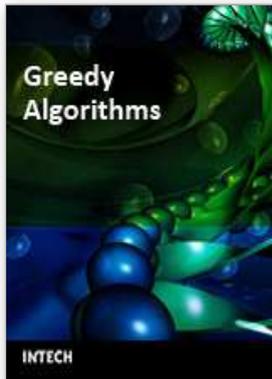
The parallel multi-user bit-load algorithm proposed in this research work reduced the number of iterations to allocated bits and power to all users. However, the computational load was still quite big even under the improved algorithm. Further improvements are required. For example, the idea of bit removal algorithm in (Sonalker & Shively, 1998) could be applied in the multi-user bit loading. In long-length loops, many high-frequency subchannels have no chance to get bits assigned to them. The better way is to turn off these subchannels before the algorithm starts. The computational load could be reduced even more when those "bad" subchannels are eliminated from the bit-loading process.

It is implied in the multi-user bit loading that a central process unit must exist in the network. The central unit possesses the channel information of all users, such as channel transfer function and crosstalk transfer function. The bit-loading algorithm runs on the central unit and the loading scheme is then distributed to all transmitters in the same cable. Therefore, the channel and crosstalk estimation need to be added into the multi-user bit-loading system (Cioffi et al., 2001) and (Zeng et al., 2001).

8. Reference

- Akujuobi, C. M. & Shen, J. (2006). A New Parallel Greedy Bit-Loading Algorithm With Fairness for Multi-Users in a DMT System, *IEEE Transactions on Communications*, Vol. 54, No. 8, August 2006
- ANSI Std. T1.417 (2001). Spectrum Management for Loop Transmission Systems, *American National Standard Institute*

- Campello, J. (1998). Optimal Discrete Bit Loading for Multicarrier Modulation Systems, *Proceedings of IEEE International Symposium on Information Theory*, pp. 193, 1998
- Campello, J. (1999). Practical Bit Loading for DMT, *IEEE International Conference on Communications*, Vol. 2, pp 801-805, June 1999
- Cioffi, J. M.; Aldana, C.; Ekbal, A. ; Fang, J.; Ginis, G.; Lee, J.; Salvekar, A.; Yu, W.; and Zeng, C. (2001). Channel Identification with Dynamic Spectrum Management, *T1E1.4 Spectrum Management II Project*, 2001-147, May 2001
- Cioffi, J. M. (2001). Proposed Scope and Mission for Dynamic Spectrum Management, *T1E1.4 Spectrum Management Project*, 2001-188R4, November 2001
- Cioffi, J. M.; Yu, W.; Chung, S. T.; and Ginis, G. (2001). Iterative Water-filling for DSM, *T1E1.4 Spectrum Management Project*, 2001-200R5, November 2001
- Chow, P. S.; Cioffi, J. M.; Bingham, J. A. C. (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, February/March/April 1995
- Chow, P. S.; Cioffi, J. M. (1995). Method and Apparatus for Adaptive, Variable Bandwidth High-Speed Data Transmission of a Multicarrier Signal Over Digital Subscriber Lines, *U.S. Patent 5,479,447* (December 1995)
- Cover, T. M. & Thomas, J. A. (1991). Elements of Information Theory, *John Wiley & Sons*, 1991
- Gallager, R. G. (1968). Information Theory and Reliable Communication, *John Wiley*, New York, 1968
- Hughes-Hartogs, D. (1987-1989). Ensemble Modem Structure for Imperfect Transmission Media, *U.S. Patent 4,679,227* (July 1987), 4,731,816 (March 1988), and 4,883,706 (May 1989)
- Lee, J.; Sonalkar, R. V. and Cioffi, J. M. (2002). Multi-user discrete bit-loading for DMT-based DSL Systems, *IEEE Global Telecommunications Conference*, Vol. 2, pp. 1259-1263, November 2002
- Leke, A. and Cioffi, J. M. (1997). A Maximum Rate Loading Algorithm for Discrete Multitone Modulation Systems, *IEEE Global Telecommunications Conference*, Vol. 3, pp. 1514-1518, November 1997
- Marler, R. T. and Arora, J. S. (2003). Review of Multi-Objective Optimization Concepts and Algorithms for Engineering, University of Iowa Optimal Design Laboratory, *Technical Report Number ODL-01.03*, 2003
- Shannon, C. E. (1948). A Mathematical Theory of Communication, *The Bell System Technical Journal*, Vol. 27, pp.379-423, 623-656, 1948
- Sonalkar, R. V. & Shively, R. R. (1998). An Efficient Bit-Loading Algorithm for DM Applications, *IEEE Global Telecommunications Conference*, Vol. 5, pp 8-12, November 1998
- Starr, T.; Cioffi, J. M.; Silverman, P. J. (1999). Understanding Digital Subscriber Line Technology, *Prentice Hall*, New York.
- Tu, J. C.; and Cioffi, J. M. (1990). A Loading Algorithm for the Concatenation of Coset Codes with Multichannel Modulation Methods, *IEEE Global Telecommunications Conference*, pp. 1183-1187, December 1990
- Valenti, C. (2002). NEXT and FEXT Models for Twisted-Pair North American Loop Plant, *IEEE Selected Areas in Communications Journal*, Vol. 20, No. 5, June 2002, pp. 893-900
- Yu, W.; Cioffi, J. M.; (2001). On Constant Power Water-filling, *Proc. IEEE International Conference on Communications*, Vol. 6, pp. 1665-1669, June 2001
- Yu, W.; Ginis, G.; Cioffi, J. M. (2002). Distributed Multiuser Power Control for Digital Subscriber Lines, *IEEE Journal of Selected Areas in Communications*, Vol. 20, No. 5, pp. 1105-1115, June 2002
- Zeng, C.; Aldana, C.; Salvekar, A. and Cioffi, J. M. (2001). Crosstalk Identification in xDSL Systems, *IEEE Journal of Selected Areas in Communications*, , Vol. 19, No. 8, pp. 1488-1496, August 2001



Greedy Algorithms

Edited by Witold Bednorz

ISBN 978-953-7619-27-5

Hard cover, 586 pages

Publisher InTech

Published online 01, November, 2008

Published in print edition November, 2008

Each chapter comprises a separate study on some optimization problem giving both an introductory look into the theory the problem comes from and some new developments invented by author(s). Usually some elementary knowledge is assumed, yet all the required facts are quoted mostly in examples, remarks or theorems.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Cajetan M. Akujuobi and Jie Shen (2008). Efficient Multi-User Parallel Greedy Bit-Loading Algorithm with Fairness Control For DMT Systems, *Greedy Algorithms*, Witold Bednorz (Ed.), ISBN: 978-953-7619-27-5, InTech, Available from: http://www.intechopen.com/books/greedy_algorithms/efficient_multi-user_parallel_greedy_bit-loading_algorithm_with_fairness_control_for_dmt_systems

INTECH

open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2008 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.