Reprint

# A New Computationally Efficient Discrete Bit-Loading Algorithm for DMT Applications

*N. Papandreou and Th. Antonakopoulos*

IEEE Transactions on Communications

VOL. 53 NO. 5, MAY 2005, pp. 785-789

# A New Computationally Efficient Discrete Bit-Loading Algorithm for DMT Applications

Nikolaos Papandreou, *Member, IEEE,* and Theodore Antonakopoulos, *Senior Member, IEEE*

***Abstract*—This letter presents a new bit-loading algorithm for discrete multitone systems that converges faster to the same bit allocation as the optimal discrete bit-filling and bit-removal methods. The algorithm exploits the differences between the subchannel gain-to-noise ratios in order to determine an initial bit allocation and then performs a multiple-bits loading procedure for achieving the requested target rate. Numerical results using asymmetric digital subscriber test loops demonstrate the computational efficiency of the proposed algorithm.**

***Index Terms*—Bit loading, digital subscriber line (DSL), discrete multitone (DMT), multicarrier communications.**

## I. Introduction

DISCRETE MULTITONE (DMT) [1] has been adopted as the modulation scheme for the asymmetric digital subscriber line (ADSL) [2] and the very-high-bit-rate DSL (VDSL) [3] technologies. In multicarrier systems, a *loading algorithm* is used to allocate bits and power to the subchannels. This task is a constrained optimization problem and two cases are of interest, namely, *margin maximization* and *rate maximization* [4].

Margin maximization is equivalent to power minimization subject to a fixed target rate, while rate maximization predefines a given minimum margin. In both cases, system specifications [2], [5] impose additional constraints including total power budget, power spectral density (PSD) mask, maximum bit error rate (BER), integer bit assignments, and maximum size of the embedded QAM constellations. There are several single-user bit-loading algorithms for DMT-based modems, however, not all of them are optimal under the integer bit constraint, and not all of the above constraints are encountered in the problem formulations.

In [6]–[8], the bit-loading solution is generally noninteger; therefore, a suboptimal integer rounded bit allocation is provided. In [9]–[11], the discrete bit-loading problem is addressed using an iterative bit-filling or bit-removal procedure based on a subchannel-cost criterion for a total power budget constraint. A discrete solution is also provided in [12] using the Lagrange approach that involves a lookup-table search method. In [13]–[15], the PSD mask limitation is introduced and the greedy bit-filling and bit-removal algorithms are proven to be optimal for discrete bit loading.

Paper approved by S. K. Wilson, Editor for Multicarrier Modulation of the IEEE Communications Society. Manuscript received March 31, 2004; revised September 1, 2004. This work was supported in part by ATMEL MMC in the framework of project "Aliakmon." This work was presented in part at the IEEE Communication Theory Workshop, Capri, Italy, May 2004.

The authors are with the Department of Electrical Engineering and Computers Technology, University of Patras, 26500 Rio-Patras, Greece (e-mail: npapandr@loe.ee.upatras.gr; antonako@ee.upatras.gr).

Digital Object Identifier 10.1109/TCOMM.2005.847141

In this letter, we present a computationally efficient discrete bit-loading algorithm for the single-user margin maximization problem. The new algorithm exploits the differences between the subchannel gain-to-noise ratios measured during channel training in order to calculate an initial bit allocation and then performs a multiple-bits loading procedure that converges fast to the optimal target-rate solution.

Section II provides a concise description of the discrete bit-allocation problem, while Section III presents in detail the new bit-loading algorithm. In Section IV, we present numerical results using standard ADSL loops that demonstrate the performance improvement of the new algorithm compared with the optimal greedy methods.

## II. Discrete Bit-Allocation Problem

DMT modulation decomposes the channel spectrum into a set of $N$ orthogonal subchannels. The number of bits at each subchannel is given by

$$b_i = \log_2\left(1 + \frac{P_i \cdot \mathrm{CNR}_i}{\Gamma}\right) \tag{1}$$

where $\mathrm{CNR}_i = |H_i|^2/\mathcal{N}_i$ is the gain-to-noise ratio of subchannel $i$, $P_i$ denotes the power, and $\Gamma$ is the SNR gap that is calculated according to the "gap-approximation" analysis [1] based on the target BER, the applied coding, and the system performance margin. Some useful comments on the validity limits of the "gap-approximation" can be found in [8] and [15].

The discrete loading problem is stated as follows:

$$\begin{aligned} &\text{minimize} \sum_{i=1}^{N} P_i \\ &\text{subject to} \sum_{i=1}^{N} b_i = B_{\text{target}}, \quad \sum_{i=1}^{N} P_i \le P_{\text{budget}}, \\ &\qquad 0 \le b_i \le \bar{b}_i, \quad b_i \in \mathbb{Z}_+, \quad \text{for } 1 \le i \le N \end{aligned} \tag{2}$$

where $B_{\text{target}}$ is the target rate, $P_{\text{budget}}$ is the power budget, and $\bar{b}_i$ is the maximum number of bits that can be assigned to each subchannel according to

$$\bar{b}_i = \min\left(b_{\max}, \left\lfloor \log_2\left(1 + \frac{\bar{P}_i \cdot \mathrm{CNR}_i}{\Gamma}\right)\right\rfloor\right) \tag{3}$$

where $b_{\max}$ corresponds to the maximum allowable size of the QAM constellations, and $\bar{P}_i$ is the maximum allowable power imposed by the PSD mask.

The power required to transmit one more bit in subchannel $i$ that is carrying $b_i$ bits is given by

$$\Delta P_i^{+}(b_i) = 2^{b_i}\frac{\Gamma}{\mathrm{CNR}_i}, \quad 0 \le b_i < \bar{b}_i \tag{4}$$

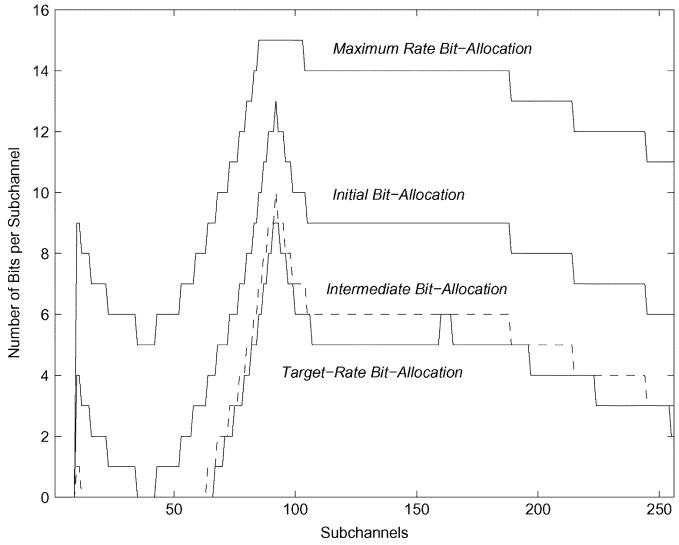


Fig. 1. Example of bit-loading profiles and bounds.

while the power saved by removing one bit is given by

$$\Delta P_i^-(b_i) = 2^{(b_i-1)}\frac{\Gamma}{\mathrm{CNR}_i}, \quad 0 < b_i \le \bar{b}_i. \tag{5}$$

For a fixed target rate, the bit-filling algorithm starts from an initial all-zeros bit allocation ($b_i = 0$ for $1 \le i \le N$) and adds one bit at a time to the subchannel that requires the minimum additional power until the target rate is achieved. On the other hand, the bit-removal algorithm starts from an initial maximum bit allocation ($b_i = \bar{b}_i$ for $1 \le i \le N$) and removes one bit at a time from the subchannel that saves the maximum power until the target rate is achieved.

The logarithmic rate expression in (1) is a strictly increasing and concave function of $P_i$ and vanishes at $P_i = 0$. These conditions guarantee the optimality of the above greedy bit-filling and bit-removal methods [13]–[15]. Moreover, it can be proved that both methods result to the same bit allocation. However, the computational load associated with each algorithm depends mainly on the requested target rate. If $B_{\mathrm{target}}$ is closer to the rate achieved by the $\bar{b}_i$ allocation, then bit-removal converges faster.

## III. New Bit-Loading Algorithm

For each loop type and for certain noise conditions, there is a maximum achievable data rate due to the system constraints. This maximum data rate corresponds to the $\bar{b}_i$ allocation and is shown as the *maximum-rate bit-allocation*[1] curve in the example of Fig. 1. For a given target rate, there exists a unique bit-allocation profile, called *target-rate bit-allocation*, that corresponds to the optimum discrete solution to (2). Bit-filling and bit-removal converge to the *target-rate bit-allocation* curve in Fig. 1 starting either from the all-zeros or from the *maximum-rate bit-allocation* curves, respectively.

[1]If the total power of (3), for $1 \le i \le N$, exceeds $P_{\mathrm{budget}}$, then the most power-expensive bits determined by (5) have to be removed in order to meet the power budget constraint and the new distribution $\bar{b}_i$ determines the *maximum-rate bit-allocation* and is also the solution to the discrete rate maximization problem, where any minimum margin is inherited in (3).

Our bit-loading algorithm uses a three-step procedure for converging to the same *target-rate bit-allocation* curve. Initially, the algorithm exploits the differences between the CNRs and calculates an initial bit allocation within the system constraints, shown as the *initial bit-allocation* curve (Fig. 1). Then, the algorithm compares the achieved data rate with the target rate and follows a multiple-bits insertion or removal procedure in order to approach the optimum bit distribution, with no more than a single bit difference per subchannel. This curve is denoted as *intermediate bit-allocation* (dashed line). At the final step, bit-filling or bit-removal is used in order to achieve the *target-rate bit-allocation* curve.

Initially, we examine the process of bit-filling, without considering the $\bar{b}_i$ limit. Let $i_{\max} = \arg\max\{\mathrm{CNR}_i\}$ and $i_{\min} = \arg\min\{\mathrm{CNR}_i\}$, for $1 \le i \le N$. If we examine the bit allocation on subchannels $i_{\max}$ and $m$, bit-filling loads subchannel $i_{\max}$ with $b_m = \lfloor \log_2(k_m) + 1 \rfloor$ bits before the first bit is added to subchannel $m$, where $k_m = (\mathrm{CNR}_{i_{\max}})/(\mathrm{CNR}_m)$ and $b_m$ is the integer solution of

$$\Delta P^+_{i_{\max}}(b_m - 1) \le \Delta P^+_m(0) < \Delta P^+_{i_{\max}}(b_m). \tag{6}$$

Then, any additional bits with respect to subchannels $i_{\max}$ and $m$ are successively added, starting from subchannel $m$. The above statement is valid, since for any integer $x \ge 0$

$$\Delta P^+_m(x+1) \ge \Delta P^+_{i_{\max}}(b_m + x) > \Delta P^+_m(x). \tag{7}$$

Similarly, if we examine bit allocation on subchannels $i_{\max}, m$, and $n$, where $\mathrm{CNR}_m > \mathrm{CNR}_n$, then bit-filling adds $b_n = \lfloor \log_2(k_n) + 1 \rfloor \ge b_m$ bits to subchannel $i_{\max}$ before the first bit is added to subchannel $n$, where $k_n = (\mathrm{CNR}_{i_{\max}})/(\mathrm{CNR}_n)$, while subchannel $m$ is filled with $b_n - b_m$ bits. The last remark is straightforward by setting $x = b_n - b_m$ in (7). Then, any additional bits with respect to subchannels $i_{\max}, m$, and $n$ are inserted based on the minimum required power, so that, after every three allocations on subchannels $i_{\max}, m$, and $n$, they are filled with one more bit. In the following subsections, the above remarks are extended to all $N$ subchannels.

### A. Initial Bit Allocation

Let $k_i = (\mathrm{CNR}_{i_{\max}})/(\mathrm{CNR}_i)$, for $1 \le i \le N$. The first bit is added to subchannel $i_{\min}$ after the following allocation:

$$b'_i = \begin{cases} \lfloor \log_2(k_{i_{\min}}) + 1 \rfloor, & i = i_{\max} \\ \lfloor \log_2(k_{i_{\min}}) \rfloor - \lfloor \log_2(k_i) \rfloor, & \text{otherwise.} \end{cases} \tag{8}$$

Profile $b'_i$ depends only on the CNRs and is a loop-representative bit allocation under the power minimization goal. However, (8) does not include any information about the corresponding transmit power. As a result, $b'_i$ might be invalid under constraint $\bar{b}_i$.

Initially, we calculate the $b'_i$ profile and, if the $\bar{b}_i$ bit mask is violated, we downshift profile $b'_i$ by $\triangle b = \max\{b'_i - \bar{b}_i\}$. In this case, the new bit profile does not violate the $\bar{b}_i$ bit mask, but it may contain negative values in some subchannels. These values do not participate in the data rate calculations and their

physical interpretation is explained in the next subsection. In the Appendix, the following proposition is proven.

*Proposition:* $b'_{i_{\max}} - \bar{b}_{i_{\max}} = \max\{b'_i - \bar{b}_i\}, \forall i : 1 \le i \le N.$

Therefore, the *initial bit-allocation* profile of the presented bit-loading algorithm is given by

$$b_i^{\text{init}} = \begin{cases} b'_i - \triangle b, & \text{if } b'_{i_{\max}} > \bar{b}_{i_{\max}} \\ b'_i, & \text{otherwise.} \end{cases} \tag{9}$$

The above bit allocation is optimum under the power minimization goal.

### B. Intermediate Bit Allocation

Using $b_i = b_i^{\text{init}}$ for $1 \le i \le N$, as the loop's initial bit profile, we have either to add or remove bits in order to achieve the target rate. In the special case, where the target rate equals the data rate achieved with the initial profile, no further processing is performed. Denoting $B = \sum_{i=1}^{N} \max\{0, b_i\}$, we distinguish two cases:

1) $B < B_{\text{target}}$: We have to add bits.
   a) Calculate $N^*$, the population of nonnegative elements in $\mathcal{I} = \{i : b_i < \bar{b}_i, 1 \le i \le N\}$.
   b) Calculate $b_1^* = \lfloor (B_{\text{target}} - B)/(N^*) \rfloor$.
   c) If $\mathcal{I}$ contains negative elements, then calculate $b_2^* = \min\{|b_i|\}$ for all $b_i < 0$ and set $a = \min\{b_1^*, b_2^*\}$, else set $a = b_1^*$.
   d) If $a \neq 0$, then proceed to step e), else $b_i$ determines the *intermediate bit-allocation* profile.
   e) Add $a$ bits to all subchannels in $\mathcal{I}$, then upper-bound $b_i$ with $\bar{b}_i$ and update $B$.
   f) If $B = B_{\text{target}}$, then the allocation $\max\{0, b_i\}$ determines the *target-rate bit-allocation* profile, else proceed to step a).
2) $B > B_{\text{target}}$: We have to remove bits.
   a) Calculate $N^*$, the population of $\mathcal{I} = \{i : b_i > 0, 1 \le i \le N\}$.
   b) Calculate $a = \lfloor (B - B_{\text{target}})/(N^*) \rfloor$.
   c) If $a \neq 0$, then proceed to step d), else $b_i$ determines the *intermediate bit-allocation* profile.
   d) Remove $a$ bits from all subchannels in $\mathcal{I}$ and update $B$.
   e) If $B = B_{\text{target}}$, then the allocation $\max\{0, b_i\}$ determines the *target-rate bit-allocation* profile, else proceed to step a).

When $B > B_{\text{target}}$, we can perform multiple-bits removal from all subchannels, so that, after every $a$ bits removal, each subchannel contains $\max\{b_i - a, 0\}$ bits. When $B < B_{\text{target}}$, the following procedure results to proper bit allocation not exceeding $B_{\text{target}}$. If $b'_{i_{\max}} \le \bar{b}_{i_{\max}}$, we can perform multiple-bits addition to all subchannels, so that after every $a$ bits addition, each subchannel contains $\min\{b_i + a, \bar{b}_i\}$ bits. However, if $b'_{i_{\max}} > \bar{b}_{i_{\max}}$, then, according to (9), $b_i^{\text{init}}$ distribution may contain negative values in some subchannels. Let $b_j \le b_i < 0$. Then, we need to add $b_i$ bits to all subchannels before we can add the first bit in subchannel $i$, and at that time subchannel $j$ is assigned with $b_j - b_i$ bits. Such restrictions determine the value of $a$, when multiple-bits addition is performed.

### C. Final Bit Allocation

The final step for achieving the target rate is to use either bit-filling or bit-removal for the nonallocated bits starting from the *intermediate bit-allocation* profile, which corresponds to a total data rate of $B$ bits. Again, we distinguish two cases.

1) $B < B_{\text{target}}$: Perform bit-filling for the remaining $B_{\text{target}} - B$ bits in the set of subchannels $\mathcal{I}_+ = \{i : \bar{b}_i > b_i \ge 0, 1 \le i \le N\}$.
2) $B > B_{\text{target}}$: Perform bit-removal for the remaining $B - B_{\text{target}}$ bits in the set of subchannels $\mathcal{I} = \{i : b_i > 0, 1 \le i \le N\}$.

In both cases, the final bit allocation is given by $\max\{0, b_i\}$ and corresponds to the minimum total power under the system constraints. The difference between the power budget and the total power is exploited for increasing the system margin.

The new bit-loading algorithm provides the optimum discrete solution to (2). It can be proved that the characteristic bit distribution in (8) and any down or up shifted version of it, comprises an *efficient* distribution as defined by Campello in [16], also taking into account that $\bar{b}_i$ upper bounds any subchannel. Therefore, the *initial* and the *intermediate bit allocations* are also *efficient* bit distributions, while the final step of the new algorithm also results to *efficient* bit allocations until the optimum target-rate solution is achieved.

## IV. Numerical Results

In this section, we present numerical results that demonstrate the improved performance of the new bit-loading algorithm. The algorithm was evaluated using the standard ADSL crosstalk testing scenarios in Table 47 of ANSI T1.413-1995 [2]. In all cases, $-140$ dBm/Hz AWGN is considered in order to form the composite noise level.

For each testing scenario, Table I presents the relation between the target rate (in bits per DMT symbol) and the following parameters of the new algorithm: the rate achieved after the *initial bit-allocation* phase, the number of multiple additions or removals executed during the *intermediate bit-allocation* phase, and the number of bits remained to be allocated using bit-filling or bit-removal at the *final bit-allocation* phase. The numbers in parentheses correspond to the percentage of the total execution time of each phase. For each scenario, the target rate corresponds to 10%, 50%, and 90% of the loop's maximum rate, where a minimum margin of 3 dB for the mid-CSA loop and 6 dB for all other loops was included. In all cases, we have used the following constraint values: 100 mW total power budget, $-40$ dBm/Hz PSD mask, $10^{-7}$ BER, $b_{\max} = 15$ and a 40-kHz lower band edge.

In order to measure the performance of the proposed method, the new bit-loading algorithm along with the optimum bit-filling and bit-removal algorithms, which also consider the constraints described in (2), were implemented in a 900-Mflops DSP. Table I also presents the execution time of each algorithm along with the performance improvement achieved using the proposed method. In most cases, the new algorithm has much less computational complexity compared to the bit-filling and bit-removal algorithms, while it may exhibit slight performance degradation when the target rate is very low (compared to

TABLE I
EXPERIMENTAL RESULTS AND PERFORMANCE IMPROVEMENT FOR TARGET RATES EQUAL TO 10%, 50%, AND 90% OF THE LOOP'S MAXIMUM RATE

| Testing scenarios | Target rate | Algorithm's bit-allocation phases | | | Execution time (msec) | | | Performance improvement compared to | |
|---|---|---|---|---|---|---|---|---|---|
| | | *Initial* | *Intermediate* | *Final* | | | | | |
| | | Achieved rate | Multiple allocations | Remaining bits | *bit-filling* | *bit-removal* | new | *bit-filling* | *bit-removal* |
| T1.601 #7 | 45 | (90.4) | 3 (4.3) | 5 (5.3) | 11.86 | 34.66 | 14.24 | 0.83 | 2.43 |
| | 223 | 374 (76.4) | 1 (1.6) | 31 (22.0) | 24.77 | 24.55 | 16.85 | 1.47 | 1.45 |
| | 401 | (81.9) | 0 (1.7) | 27 (16.4) | 37.23 | 13.76 | 15.72 | 2.37 | 0.87 |
| T1.601 #13 | 59 | (83.0) | 2 (2.7) | 28 (14.3) | 13.42 | 44.62 | 16.17 | 0.83 | 2.76 |
| | 297 | 526 (63.1) | 1 (1.3) | 83 (35.6) | 30.63 | 30.71 | 21.28 | 1.44 | 1.44 |
| | 534 | (88.0) | 0 (1.6) | 8 (10.4) | 47.25 | 15.85 | 15.20 | 3.11 | 1.04 |
| CSA #4 | 171 | (52.2) | 1 (1.1) | 165 (46.7) | 23.37 | 125.56 | 29.10 | 0.81 | 4.31 |
| | 856 | 786 (60.5) | 0 (0.6) | 70 (38.9) | 72.95 | 78.25 | 25.11 | 2.90 | 3.11 |
| | 1540 | (74.8) | 1 (2.4) | 13 (22.7) | 122.20 | 28.20 | 20.30 | 6.02 | 1.39 |
| CSA #6 | 151 | (84.0) | 3 (3.8) | 6 (12.2) | 21.97 | 110.55 | 18.07 | 1.22 | 6.12 |
| | 754 | 992 (74.2) | 1 (1.5) | 16 (24.3) | 65.57 | 69.71 | 20.47 | 3.20 | 3.41 |
| | 1357 | (53.1) | 1 (1.8) | 118 (45.1) | 108.74 | 26.74 | 28.60 | 3.80 | 0.94 |
| CSA #7 | 178 | (60.6) | 2 (2.0) | 118 (37.4) | 23.88 | 127.32 | 25.08 | 0.95 | 5.07 |
| | 889 | 1132 (75.8) | 1 (1.6) | 4 (22.6) | 75.36 | 80.11 | 20.06 | 3.76 | 4.00 |
| | 1600 | (43.9) | 1 (1.5) | 221 (54.6) | 126.68 | 28.83 | 34.63 | 3.66 | 0.83 |
| mid-CSA | 185 | (81.4) | 2 (2.3) | 38 (16.3) | 24.32 | 123.75 | 18.75 | 1.30 | 6.60 |
| | 924 | 601 (61.1) | 2 (4.6) | 96 (34.3) | 78.03 | 80.50 | 24.96 | 3.13 | 3.22 |
| | 1662 | (68.7) | 7 (14.6) | 2 (16.7) | 133.93 | 29.66 | 22.22 | 6.03 | 1.33 |

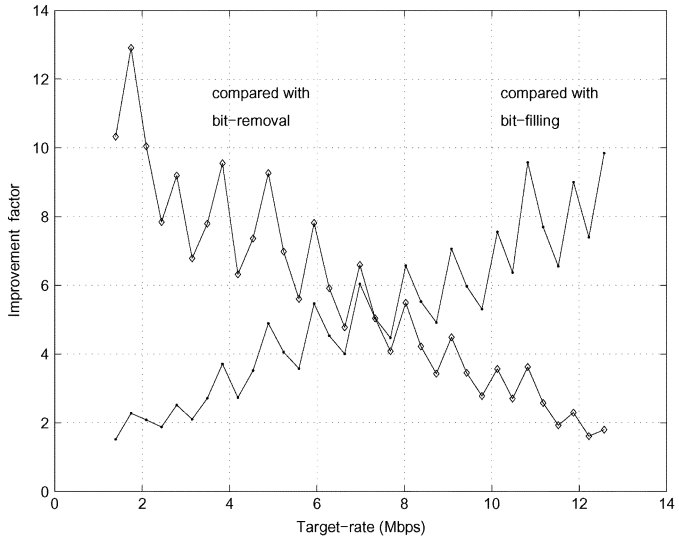


Fig. 2. Computational complexity improvement factor for the mid-CSA loop with −140-dBm/Hz AWGN.

the bit-filling algorithm) or approximates the maximum rate (compared to the bit-removal algorithm). It has to be noted that the calculation of the incremental energies of the bit-filling algorithm are based on the recursive relationship in [15], and a similar relationship is used for the bit-removal algorithm.

Fig. 2 presents the improvement factor (the ratio of the corresponding execution times) regarding the computational complexity of the proposed algorithm compared with the bit-filling and the bit-removal methods for the case of the mid-CSA loop with −140-dBm/Hz AWGN and target rates from 10% to 90% of the loop's maximum rate. Both curves have a sawtooth shape due to the different values of the number of bits remained to be allocated at the *final bit-allocation* phase. This number varies from 0 to $N-1$ and generally, for a given target rate, it equals to the modulo operation between the number of bits that need to be added (or removed) after every multiple-bits allocation during the *intermediate bit-allocation* phase, and the number

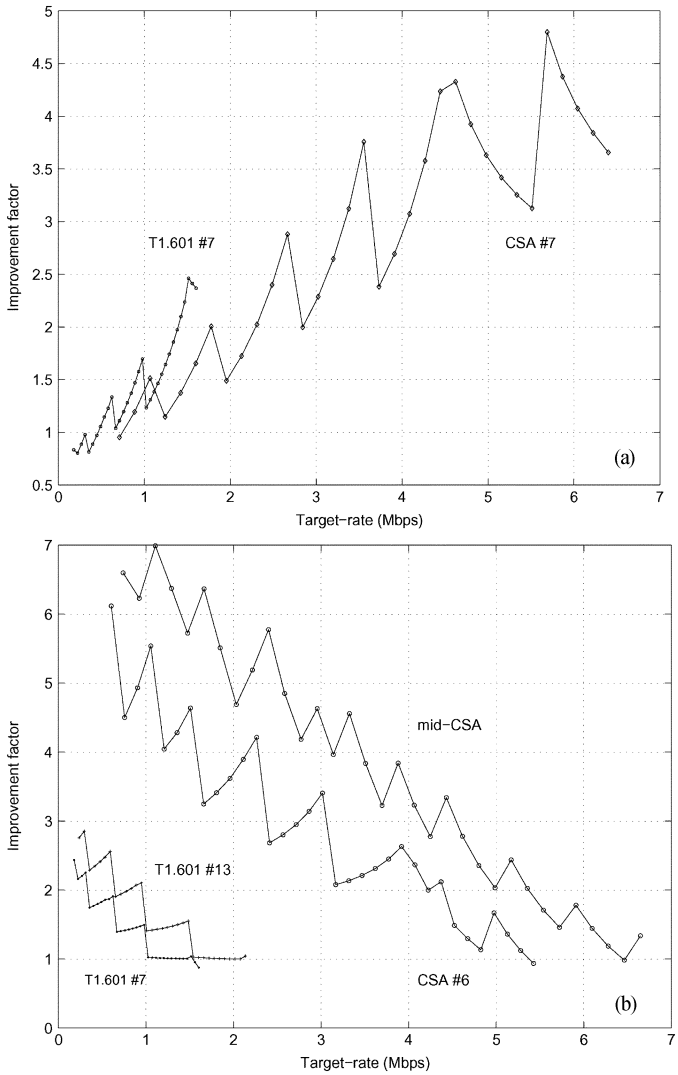


Fig. 3. Computational complexity improvement factor of the presented algorithm compared with (a) the bit-filling and (b) the bit-removal algorithms for the testing scenarios of Table I.

of subchannels that have to accept (or discard) bits. Fig. 3 also demonstrates the computational complexity improvement that can be achieved using the proposed algorithm compared with the bit-filling and the bit-removal algorithms, respectively, for various testing scenarios.

## V. Conclusion

We have presented a new discrete bit-loading algorithm that results in less computational complexity compared to the optimal bit-filling and bit-removal algorithms. The new algorithm exploits the subchannel gain-to-noise ratio levels in order to calculate a loop representative bit-allocation profile and then uses a multiple-bits insertion or removal procedure that results in faster convergence to the target rate. Numerical results showed the improved performance of the new algorithm using standard ADSL test loops.

## Appendix

We define $f_i = b'_i - \bar{b}_i, i \in \mathcal{S} = \{i : 1 \le i \le N\}$, where $b'_i$ is given in (8) and $\bar{b}_i = \min\{b_{\max}, \lfloor \log_2(1+\zeta \cdot k_i^{-1}) \rfloor\}$, where $\zeta = (\bar{P}_i \cdot \mathrm{CNR}_{i_{\max}})/(\Gamma)$ and $k_i = (\mathrm{CNR}_{i_{\max}})/(\mathrm{CNR}_i)$. We have to prove that $f_{i_{\max}} - f_i \ge 0, \forall i \in \mathcal{S}$. Using (8), we can write

$$f_{i_{\max}} - f_i = 1 + \lfloor \log_2(k_i) \rfloor - \left(\bar{b}_{i_{\max}} - \bar{b}_i\right), \quad i \ne i_{\max}. \tag{10}$$

Concerning $\bar{b}_i$, we distinguish two cases:

1) $\bar{b}_i = \lfloor \log_2(1 + \zeta \cdot k_i^{-1}) \rfloor, \forall i \in \mathcal{S}$. In this case, (10) becomes

$$f_{i_{\max}} - f_i = 1 + \lfloor \log_2(k_i) \rfloor - \lfloor \log_2(1+\zeta) \rfloor + \lfloor \log_2(1+\zeta \cdot k_i^{-1}) \rfloor, \quad i \ne i_{\max}. \tag{11}$$

Since $k_i + \zeta > k_i, \forall i \in \mathcal{S}$, we can write

$$\lfloor \log_2\left(1+\zeta \cdot k_i^{-1}\right) \rfloor = \lfloor \log_2(k_i+\zeta) \rfloor - \lfloor \log_2(k_i) \rfloor - \varepsilon_i \tag{12}$$

where $\varepsilon_i = 1$ or 0. Combining (11) and (12), and since $k_i \ge 1, \forall i \in \mathcal{S}$, we have that $f_{i_{\max}} - f_i \ge 0, \forall i \in \mathcal{S}$. Note that (11) was shown to be positive irrespective of the $b_{\max}$ value. This remark is used in the next case.

2) $\bar{b}_i = b_{\max}, \forall i \in \mathcal{S}_1$ and $\bar{b}_i = \lfloor \log_2(1+\zeta \cdot k_i^{-1}) \rfloor < b_{\max}$, $\forall i \in \mathcal{S}_2$, where $\mathcal{S}_1 \cap \mathcal{S}_2 = \emptyset$ and $\mathcal{S}_1 \cup \mathcal{S}_2 = \mathcal{S}$.
It can be verified that $\bar{b}_i$ is an increasing function with respect to $\mathrm{CNR}_i$ over $\mathcal{S}$. Therefore, $i_{\max} \in \mathcal{S}_1$ and $\forall i \in \mathcal{S}$ the following relation holds:

$$\lfloor \log_2(1+\zeta) \rfloor - \lfloor \log_2\left(1+\zeta \cdot k_i^{-1}\right) \rfloor \ge \bar{b}_{i_{\max}} - \bar{b}_i. \tag{13}$$

The first part of (13) appears in (11), which was proven to be positive. Therefore, by combining them, we get that $f_{i_{\max}} - f_i \ge 0, \forall i \in \mathcal{S}$.

In both cases, we have shown that $f_{i_{\max}} = \max\{f_i\}, \forall i \in \mathcal{S}$.

## References


[1] *A Multicarrier Primer. ANSI Contribution T1E1.4/91-157*, Nov. 1991.

[2] *Asymmetric Digital Subscriber Line (ADSL) Metallic Interface*. ANSI Standard Tl.413-1995.

[3] *Very-High-Bit-Rate Digital Subscriber Line (VDSL) Metallic Interface: Part 1: Functional Requirements and Common Specification. Part 3: Technical Specification of a Multi-Carrier Modulation Transceiver*. ANSI Contribution T1E1.4/2000-013R1, 2000-009R3.

[4] T. Starr, J. M. Cioffi, and P. J. Silverman, *Understanding Digital Subscriber Line Technology*. Upper Saddle River, NJ: Prentice-Hall, 1999.

[5] *Spectrum Management for Loop Transmission Systems*. ANSI Standard T1.417-2001.

[6] P. S. Chow, J. M. Cioffi, and J. A. C. Bingham, "A practical discrete multitone transceiver loading algorithm for data transmission over spectrally shaped channels," *IEEE Trans. Commun.*, vol. 43, no. 2–4, pp. 773–775, Feb./Mar./Apr. 1995.

[7] A. Leke and J. M. Cioffi, "A maximum rate loading algorithm for discrete multitone modulation systems," in *Proc. IEEE Globecom*, vol. 3, Nov. 1997, pp. 1514–1518.

[8] E. Baccarelli, A. Fasano, and M. Biagi, "Novel efficient bit-loading algorithms for peak-energy-limited ADSL-type multicarrier systems," *IEEE Trans. Signal Process.*, vol. 50, no. 5, pp. 1237–1247, May 2002.

[9] D. Hughes-Hartogs, "Ensemble Modem Structure for Imperfect Transmission Media," U.S. Patents 4 679 227 (Jul. 1987), 4 731 816 (Mar. 1988), and 4 833 796 (May 1989).

[10] J. Campello, "Practical bit loading for DMT," in *Proc. IEEE ICC*, vol. 2, Jun. 1999, pp. 801–805.

[11] R. V. Sonalkar and R. R. Shively, "An efficient bit-loading algorithm for DMT applications," *IEEE Commun. Lett.*, vol. 4, no. 3, pp. 80–82, Mar. 2000.

[12] B. S. Krongold, K. Ramchandran, and D. L. Jones, "Computationally efficient optimal power allocation algorithms for multicarrier communication systems," *IEEE Trans. Commun.*, vol. 48, no. 1, pp. 23–27, Jan. 2000.

[13] A. Fasano, G. Di Blasio, E. Baccarelli, and M. Biagi, "Optimal discrete bit loading for DMT based constrained multicarrier systems," in *Proc. IEEE ISIT*, Jul. 2002, p. 243.

[14] A. Fasano, "On the optimal discrete bit loading for multicarrier systems with constraints," in *Proc. IEEE VTC*, Apr. 2003, pp. 915–919.

[15] E. Baccarelli and M. Biagi, "Optimal integer bit-loading for multicarrier ADSL systems subject to spectral-compatibility limits," *Signal Process.*, vol. 84, pp. 729–741, Apr. 2004.

[16] J. Campello, "Optimal discrete bit loading for multicarrier modulation systems," in *Proc. IEEE ISIT*, Aug. 1998, p. 193.