Optimal Spectrum Management of DSL With Nonstrictly Convex Rate Region

Amir R. Forouzan, *Member, IEEE*

***Abstract*—Recently, the problem of optimal spectral balancing (OSB) for digital subscriber lines (DSL) with constrained transmit power has been solved using Lagrange's dual optimization technique and a weighted sum rate maximization (WSRM) approach. In many cases, the total power constraint is not binding. Although, this means a huge computational complexity reduction, the algorithm fails to reach certain points on the rate region (RR). In this paper, an in-depth analytical view of the WSRM approach is provided, and it is shown that when the RR is not strictly convex, the WSRM approach fails to reach certain points on the RR. Moreover, using N-dimensional geometry, a novel iterative facet dividing algorithm (IFDA) capable of reaching any point on the RR is proposed. Analytical and simulation results show that our technique is much more reliable and considerably faster than current algorithms. Moreover, it can be used for a wide range of problems which use WSRM approach, including OSB in the general case.**

***Index Terms*—Convex optimization, digital subscriber lines (DSL), dynamic spectrum management (DSM), interference channel, nonconvex optimization.**

I. Introduction

With more than 200 million customers around the world, digital subscriber lines (DSL) are well-known as a cheap and popular way to provide broadband internet services for the homes and small business. The higher penetration of DSL has increased the possibility and volume of bit-rate loss because of the increased crosstalk levels in the telephone loops. In many cases, crosstalk is the main impairment in DSL systems. Consequently, it is highly beneficial to cancel crosstalk in most networks. There are basically two classes of crosstalk mitigation schemes for DSL, spectrum balancing (SB) schemes and vector coding (VC) schemes. VC schemes are usually much more effective than SB schemes. However, they require signal level coordination among users, which is not possible in many situations, e.g., cabinet/ remote-terminal deployments. Nevertheless, in certain scenarios, SB can be used to improve the performance of DSL systems significantly.

To limit the crosstalk power, DSL systems are required to obey regulatory power spectral density (PSD) masks. This guarantees a minimum achievable bit-rate for DSL systems in the network, even if they belong to different service providers. It also prevents destructive competition among uncoordinated resource-hungry modems and distributes the network resources more evenly among users. This regulation is called static spectrum management (SSM) and must be observed in current and future DSL systems.

An optimal spectrum balancing (OSB) algorithm for discrete multitone (DMT) DSL using Lagrange optimization technique has been proposed in [1]. OSB uses a weighted sum rate maximization (WSRM) approach to solve this problem, and by iterating through different weight factors, it finds an optimal set of weight factors which satisfy users' bit-rate requirements. It also takes into account a maximum output power (P_{AFE}) constraint for the analog front end (AFE) of each modem. OSB uses the Lagrange optimization technique in order to enforce modems' AFE power constraints.

However, in many cases the AFE maximum power constraint will be met automatically as long as the modem conforms to a PSD mask. This happens when P_{AFE} is greater than the sum of the power under the PSD mask, i.e., $P_{\mathrm{AFE}} \geq P_{\mathrm{mask}} \triangleq \int_{\mathrm{BW}} \mathrm{PSD}_{\mathrm{mask}}(f)\,df$. In these cases, the total power constraint is not binding. For example, roughly speaking, VDSL systems operate from about d.c. to 12 MHz under a $-60\,\mathrm{dBm/Hz}$ PSD mask. Ignoring frequency division duplexing for upstream (US) and downstream (DS) bounds, the maximum transmit power under this mask is 12 mW, which can be easily met by a modem's AFE using the available technology. In this paper, we call such a system a Mask Limited Aggregate transmit Power (MLAP) DSL. As we show later using computer simulations, OSB cannot reach certain points on the rate-region (RR) for MLAP DSL. This can be a major problem, because in certain scenarios, the nearest achievable point is located far away from the optimum point. This phenomenon has been reported by other researchers as well [2].

OSB has an exponential complexity in the number of users and is impractical for more than five users. Faster OSB algorithms are needed even with smaller number of users in most practical cases. That is because broadband traffic is bursty and the desired users' bit rate requirements change very rapidly over time. The DSL channel and the noise environment also change over time, e.g., due to temperature changes or changes in the transmitted power of alien crosstalkers [3]. Therefore, it is highly desirable to reduce the number of iterations needed by OSB to tune the weight factors.

In this paper, we comprehensively analyze OSB for MLAP DSL, which enables us to understand why a WSRM approach cannot solve the OSB problem for these systems. We show that WSRM is incapable of reaching some points on the boundary of the RR, when the RR is not strictly convex. Next, we propose a package of algorithms capable of reaching any point on RR

Manuscript received July 04, 2008; accepted January 30, 2009. First published March 10, 2009; current version published June 17, 2009. The associate editor coordinating the review of this paper and approving it for publication was Dr. Xiadong Cai.

The author is with the Department of Electrical Engineering (ESAT-SCD), Katholieke Universiteit Leuven, 3001 Leuven, Belgium (e-mail: amir.forouzan@esat.kuleuven.be).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2009.2016878

1053-587X/$25.00 © 2009 IEEE

Authorized licensed use limited to: Katholieke Universiteit Leuven. Downloaded on June 22, 2009 at 05:18 from IEEE Xplore. Restrictions apply.

very closely, even when the RR is not convex. We will then analyze the computational complexity of our algorithms to show that our technique is much faster than current methods. Our approach can potentially be used in any problem where we use the WSRM approach to find certain points on the RR. In the context of dynamic spectrum management for DSL, this includes OSB in the general case [1], several suboptimal SB techniques based on the WSRM approach [10]–[16], partial crosstalk cancelation (PCC) for DSL [3], and joint SB and PCC for DSL [17]–[19].

II. System Description and Problem Definition

We assume DMT modulation with K tones and N users with synchronized signals at the receiver side. In this case, the number of bits that can be loaded to tone k of user n is calculated by

$$b_k^{(n)} = \min\left\{ b_{\max}, \left\lfloor \log_2\left(1+\Gamma^{-1}\mathrm{SNR}_k^{(n)}\right)\right\rfloor \right\} \quad (1)$$

where $b_{\max}$ is the maximum number of bits that can be loaded to each tone, $\lfloor x \rfloor$ indicates the greatest integer equal to or smaller than x, $\mathrm{SNR}_k^{(n)}$ is the signal-to-noise power ratio (SNR) on tone k for user n, and Γ is the SNR gap. The value of $b_{\max}$ is in the range 8 to 15 depending on the type of the DSL system, and

$$\mathrm{SNR}_k^{(n)} = \frac{\left|h_k^{(n,n)}\right|^2 s_k^{(n)}}{\sigma_k^{(n)} + \sum\limits_{\substack{m=1;\\ m\neq n}}^{N} \left|h_k^{(n,m)}\right|^2 s_k^{(m)}} \quad (2)$$

where $s_k^{(n)}$ is the transmit power spectral density (PSD) of user n at tone k, $h_k^{(n,m)}$ is the channel transfer function from transmitter m to receiver n at tone k, and $\sigma_k^{(n)}$ is the noise PSD at tone k of user n. Assuming frequency division signalling for upstream (US) and downstream (DS) signals, we have ignored near-end crosstalk in (2). The transmit power at each tone k is bounded by a regulatory PSD mask denoted by $s_{k,\mathrm{mask}}^{(n)}$, that is $s_k^{(n)} \leq s_{k,\mathrm{mask}}^{(n)}$.

Let us define $\boldsymbol{b}_k = \left[b_k^{(1)}, b_k^{(2)}, \ldots, b_k^{(N)}\right]^T$ and $\boldsymbol{s}_k = \left[s_k^{(1)}, s_k^{(2)}, \ldots, s_k^{(N)}\right]^T$. Using (1) and (2), $\boldsymbol{s}_k$ can be calculated as a function of $\boldsymbol{b}_k$ by [1]:

$$\boldsymbol{s}_k = (\boldsymbol{D}_k - \Gamma \boldsymbol{B}_k \boldsymbol{C}_k)^{-1} \Gamma \boldsymbol{B}_k \boldsymbol{\sigma}_k \quad (3)$$

where $\boldsymbol{D}_k = \mathrm{diag}\left\{ \left|h_k^{(1,1)}\right|^2, \left|h_k^{(2,2)}\right|^2, \ldots, \left|h_k^{(N,N)}\right|^2 \right\}$, $\boldsymbol{B}_k = \mathrm{diag}\{2^{\boldsymbol{b}_k}\} - \boldsymbol{I} = \mathrm{diag}\{2^{b_k^{(1)}}-1, 2^{b_k^{(2)}}-1, \ldots, 2^{b_k^{(N)}}-1\}$, $\boldsymbol{C}_k = \boldsymbol{G}_k - \boldsymbol{D}_k$ ($[\boldsymbol{G}_k]_{n,m} = \left|h_k^{(n,m)}\right|^2$), and $\boldsymbol{\sigma}_k = \left[\sigma_k^{(1)}, \sigma_k^{(2)}, \ldots, \sigma_k^{(N)}\right]^T$. Vector $\boldsymbol{b}_k$ is called an achievable bit-loading vector (ABLV) for tone k, if the calculated PSD values in (3) are nonnegative and smaller than the regulatory PSD mask, i.e., $0 \leq s_k^n \leq s_{k,\mathrm{mask}}^n$; for $1 \leq n \leq N$. In this paper, we use Φ_k to denote the set of all ABLVs on tone k. In general, Φ_k is not a convex set, however, if $\boldsymbol{v} = [v_1, v_2, \ldots, v_N]^T \in \Phi_k$, then $\boldsymbol{u} = [u_1, u_2, \ldots, u_N]^T \in \Phi_k$, where $0 \leq u_n \leq v_n$ for n from 1 to N.

The aggregate transmit power of each user can then be calculated using $P_{n,\mathrm{tot}} = \Delta_f \sum_{k=1}^{K} s_k^{(n)}$, where Δ_f is the tone spacing. With the output power of the nth modem's analog front-end (AFE) limited to $P_{n,\mathrm{AFE}}$, we have

$$P_{n,\mathrm{tot}} \leq P_{n,\mathrm{AFE}}. \quad (4)$$

The bit-rate of user n is calculated by $r_n = f_s \sum_k b_k^{(n)}$, where f_s is the DMT symbol rate. For simplicity, we discard f_s and consider the bit-rate of users per DMT symbol defined by

$$R_n = \sum_k b_k^{(n)} \quad (5)$$

The rate region (RR) is the set of all vectors $[R_1, R_2, \ldots, R_N]^T \in \mathbb{Z}^N$ that can be achieved under the particular channel conditions and the discussed constraints. The SB problem is defined as [1]

$$\max_{\boldsymbol{s}^{(1)}, \boldsymbol{s}^{(2)}, \ldots, \boldsymbol{s}^{(N)}} R_N \ \text{s.t.}\ R_n \geq R_{n,\mathrm{target}} \ \text{for}\ 1 \leq n \leq N-1 \quad (6)$$

where $R_{n,\mathrm{target}}$ is the target bit-rate for user n and $\boldsymbol{s}^{(n)} \triangleq \left[s_1^{(n)}, s_2^{(n)}, \ldots, s_K^{(n)}\right]^T$ denotes the PSD of user n. In this paper, we use the following alternative formulation introduced in [4], in which the bit-rate of users is maximized in the direction of the vector of target bit-rates $\boldsymbol{R}_{\mathrm{target}} = [R_{1,\mathrm{target}}, R_{2,\mathrm{target}}, \ldots, R_{N,\mathrm{target}}]^T$

$$\max_{\boldsymbol{s}^{(1)}, \boldsymbol{s}^{(2)}, \ldots, \boldsymbol{s}^{(N)}} \rho \ \text{s.t.}\ R_n \geq \rho R_{n,\mathrm{target}} \ \text{for}\ 1 \leq n \leq N \quad (7)$$

where ρ is a positive real number. While (7) always has a solution (sometimes with $\rho < 1$), (6) does not have any solution when $R_n \geq R_{n,\mathrm{target}}$ for $1 \leq n \leq N-1$ cannot be achieved even when $R_N = 0$. That is, we do not require any prior knowledge about the RR for solving (7), while we need to know the RR to some extent to solve (6). Moreover, a potential solution to (7) shares the resources fairly among users, while one for (6) treats user N differently than the other users. Since, the boundary of the RR is usually unknown, these advantages are very significant in practice. We call (7) a ray-directed (RD) SB problem.

A. OSB Using Lagrange Optimization Technique

In [1], it has been proposed that the SB problem (6) can be solved using the following weighed sum rate maximization (WSRM) approach

$$\max_{\boldsymbol{s}^{(1)}, \boldsymbol{s}^{(2)}, \ldots, \boldsymbol{s}^{(N)}} \sum_{n=1}^{N} w_n R_n \quad (8)$$

where $w_n (0 \leq w_n \leq 1)$ for $1 \leq n \leq N$ are auxiliary weight factors. To reach different points on the RR corresponding to different target bit-rates, the weight factors should be tuned appropriately. The algorithm uses Lagrange multipliers to decouple the problem into K per-tone subproblems. It finds the optimal PSD for each tone k by finding the bit-loading and PSD vector pair $(\boldsymbol{b}_k, \boldsymbol{s}_k)$ that maximize the following Lagrangian:

$$\boldsymbol{\mathcal{L}}_k = \sum_{n=1}^{N} w_n b_k^n - \sum_{n=1}^{N} \lambda_n s_k^n \quad (9)$$

Authorized licensed use limited to: Katholieke Universiteit Leuven. Downloaded on June 22, 2009 at 05:18 from IEEE Xplore. Restrictions apply.

TABLE I
SIMULATION PARAMETERS

PARAMETER	VALUE
Crosstalk model	ANSI standard 1% worst-case model [8] without considering FSAN power sum rule
Bandplan	VDSL 997 (DS: 0.138-3, 5.1-7.05 MHz, US: 3-5.1, 7.05-12 MHz) [9]
Cable type	26 AWG [8]
Noise	White noise, -140 dBm/Hz
Tone spacing, Δ_f	4.3125 kHz
Symbol rate, f_s	4 kHz
$b_{\max}$	15
Target bit error rate	10^{-7}
Modulation gap	9.75 dB
Noise margin	6 dB
Coding gain	3.75 dB
SNR gap, Γ	12.0 dB

where $\lambda_n (\lambda_n \geq 0)$ is the Lagrange multiplier for user n. The Lagrange multipliers are nonnegative and are used to satisfy users' power constraints in (4). The algorithm consists of N nested loops corresponding to N Lagrange multipliers. In the nth loop, the minimum value for λ_n satisfying (4) is determined. In the most inner loop, the optimal value of $\boldsymbol{b}_k$ that maximizes the Lagrangian in (9) is determined for all of the tones using K exhaustive searches. Moreover, to find the optimal weight factors $N-1$ other nested loops are used. Each loop corresponds to one weight factor w_n for $1 \leq n \leq N-1$. The Nth weight factor is calculated by $\sum_{n=1}^{N} w_n^2 = 1$.

For MLAP DSL, users' power constraints are satisfied automatically. The Lagrange multipliers go to zero and the Lagrangian in (9) reduces to the following weighted sum of bits (WSB)

$$\mathcal{W}_k = \sum_{n=1}^{N} w_n b_k^n. \tag{10}$$

In this case, the N nested loops for finding λ_n are omitted and the computational complexity of the algorithm reduces by a factor of 33^N. Thirty three is the number of iterations required to find the optimal value of each Lagrange multiplier with an accuracy of 1% [1].

Although maximizing $\mathcal{W}_k$ in (10) results in a point on the boundary of the RR, this does not guarantee that we can find all of the points on the boundary of the RR by changing the weight factors. Unfortunately, this condition almost always occurs for MLAP DSL. In the next section, we study a simple scenario where although the RR is convex, (8) leads to a considerably suboptimal result, because the RR is not strictly convex.

B. Case Study: When OSB is Inferior to SSM

The case under study consists of two VDSL users communicating in the upstream (US) direction over a shared 250-m cable with the simulation parameters as in Table I. Suppose we want to maximize $R = R_1 = R_2$. If we use SSM, i.e., both users transmit at the maximum power PSD mask level, the bit rate of users will be $R_1 = R_2 = 46.20\,\text{Mbps}$. As it is a symmetric scenario, for OSB we expect that setting $w_1 = w_2 = 0.5^{1/2}$ would result in the desired answer. However, we get a highly asymmetric answer of $r_1 = 6.052\,\text{Mbps}$ and $r_2 = 98.004\,\text{Mbps}$. If we set $w_1 = (0.5+\varepsilon)^{1/2}$ and $w_2 = (0.5-\varepsilon)^{1/2}$ for an arbitrary small positive real number ε, we get $r_1 = 98.004\,\text{Mbps}$ and $r_2 = 6.052\,\text{Mbps}$. In contrast with the result obtained by SSM, both of the OSB results are located on the RR, however, one of the bit-rates is considerably smaller than what we get using SSM. From the convexity of the RR [1], we know that for any point on the straight line connecting two points on the boundary of the RR (e.g., $R_1 = R_2 = 52.028$), there exists a set of PSD and bit loadings to reach it. However, no $w_1 \in [0,1]$ leads to a better result than the SSM result. Fig. 1 shows the RR achieved by OSB. This phenomenon has recently been noticed by other researchers as well [2], however, they attributed it to the inadequacies of the double precision arithmetic used in simulations. In this paper, we will explain the theoretical cause of this phenomenon and propose a technique to reach points on the RR that are much closer to the desired points.

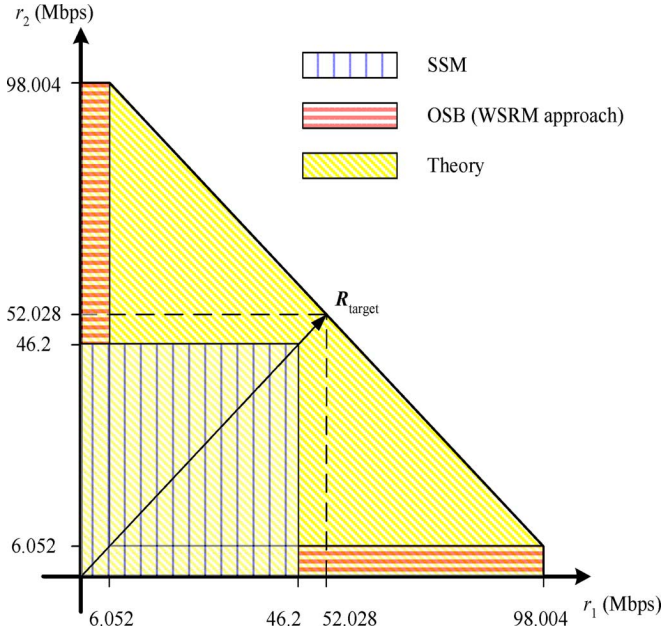

Fig. 1. Rate-region achieved by SSM, OSB, and the expected theoretical RR.

III. SOME OBSERVATIONS ON RR CALCULATION

In this section, we will provide some observations regarding RR calculation, which help us to find out the reason for suboptimality of the WSRM approach for MLAP DSL. The first step is to define the general method of RR calculation based on the multiuser bit-loadings of users in each tone. Assume Φ_k is the set of all achievable bit-loading vectors (ABLV) on tone k. Then, the RR is calculated by

$$RR = \Phi_1 \oplus \Phi_2 \oplus \cdots \oplus \Phi_K \tag{11}$$

where $\oplus$ denotes the Minkowski sum (MS) between two sets in $\mathbb{R}^N$ defined by

$$A \oplus B = \{\boldsymbol{a}+\boldsymbol{b} \,|\, \text{for all } \boldsymbol{a} \text{ and } \boldsymbol{b}; \boldsymbol{a} \in A \text{ and } \boldsymbol{b} \in B\}. \tag{12}$$

Computational complexity of the above MS is $\mathcal{O}(mn)$, where m and n are the cardinality of A and B. The cardinality of the

Authorized licensed use limited to: Katholieke Universiteit Leuven. Downloaded on June 22, 2009 at 05:18 from IEEE Xplore. Restrictions apply.

matrices Φ_k is $\mathcal{O}((b_{\max}+1)^N)$. Therefore, calculation of RR is an $\mathcal{O}((b_{\max}+1)^{NK})$ operation using (11). Fortunately, we are only interested in the boundary of the RR. The OSB algorithm in [1] uses a WSRM approach to solve this problem. In the following, we study the characteristics of this approach. In order to do this, we need to define the convex hull of the ABLV sets:

Definition: The *convex hull* (CH) of a set of points A in N dimensions is the intersection of all convex sets containing A.

From the definition, the convex hull of a set of M points $X = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_M\}$ is then given by [6]

$$\mathrm{CH}(X) = \left\{ \sum_{n=1}^{M} \rho_n \boldsymbol{p}_n \;\middle|\; \rho_n \geq 0 \text{ for all } n \text{ and } \sum_{n=1}^{M} \rho_n = 1 \right\}. \tag{13}$$

Since Φ_k consists of finite number of points in $\mathbb{R}^N$, $\mathrm{CH}(\Phi_k)$ is a convex N-dimensional (N-D) polytope. Note that the subspace $S(\boldsymbol{p}_1, \ldots, \boldsymbol{p}_N) = \left\{ \sum_{n=1}^{N} \rho_n \boldsymbol{p}_n \middle| \rho_n \geq 0; \sum_{n=1}^{N} \rho_n = 1 \right\}$ is a convex $(N-1)$-D polytope bounded to the points $\boldsymbol{p}_n$ in the N-D space. When the points $\boldsymbol{p}_n$ (for $n = 1$ to N) are vertices of $\mathrm{CH}(\Phi_k)$, S is located on the boundary of $\mathrm{CH}(\Phi_k)$ and is part of a *facet*. Each facet of any polytope A includes at least N vertices of the polytope. A facet is a subset of the hyperplane passing through its vertices. *Hyperplane* is a generalization of a plane in 3-D geometry to higher dimensions. A hyperplane in $\mathbb{R}^N$ is described by the following equation:

$$\sum_{n=1}^{N} \omega_n x_n = \eta \tag{14}$$

where ω_n and η are constants and x_n is the nth coordinate. Without any loss of generality, we assume $\eta \geq 0$ and $\sum_{n=1}^{N} \omega_n^2 = 1$. A hyperplane divides the space into two half-spaces $\sum_{n=1}^{N} \omega_n x_n \leq \eta$ and $\sum_{n=1}^{N} \omega_n x_n \geq \eta$. Alternatively, a hyperplane can generally be identified by N of its points. The $N+1$ constants ω_n and η can be calculated by solving the following set of equations:

$$\begin{cases} \boldsymbol{\omega}^T \boldsymbol{p}_m = \eta, & \text{for } 1 \leq m \leq N. \\ \sum\limits_{n=1}^{N} \omega_n^2 = 1. \end{cases} \tag{15}$$

where $\boldsymbol{\omega} = [\omega_1, \omega_2, \ldots, \omega_N]^T$ is the normal vector to the hyperplane and $\boldsymbol{p}_m = \left[p_1^{(m)}, p_2^{(m)}, \ldots, p_N^{(m)}\right]^T$ is the mth available point.

The following properties hold for $\mathrm{CH}(\Phi_k)$. The proofs are given in the Appendix .

Lemma 1: A vertex of $\mathrm{CH}(\Phi_k)$ is a member of Φ_k.

Lemma 2: The solution to the WSRM problem for tone k, i.e., $\left(b_k^1, \ldots, b_k^N\right) = \arg\max \boldsymbol{\mathcal{W}}_k$, is a vertex of $\mathrm{CH}(\Phi_k)$.

In Lemma 2, we assume that the algorithm examines the points in N nested loops. It is important to note that the lemma implies that nonvertex points of $\mathrm{CH}(\Phi_k)$ cannot be a solution to the WSRM problem, even if they maximize the WSB for a weight vector. Eight members of Φ_k are located on the outer boundary of $\mathrm{CH}(\Phi_k)$ for the example of Fig. 8. However, only the vertices $\boldsymbol{p}$, $\boldsymbol{q}$, $\boldsymbol{r}$, and $\boldsymbol{s}$ [illustrated on Fig. 8(b)] might be selected by the WSRM approach. Point $\boldsymbol{r}$ is selected when $w_1 = 1 (w_2 = 0)$, $\boldsymbol{p}$ is selected when $1 > w_1 > 0.5^{1/2} (0 < w_2 < 0.5^{1/2})$, $\boldsymbol{q}$ is selected when $0.5^{1/2} \geq w_1 > 0 (0.5^{1/2} \leq w_2 < 1)$, and $\boldsymbol{s}$ is selected when $w_1 = 0 (w_2 = 1)$.

Lemma 3: If (14) describes a hyperplane passing through a facet F on $\mathrm{CH}(\Phi_k)$, then the points belonging to Φ_k on F maximize a WSB with weight vector $[w_1, w_2, \ldots, w_N]^T = [\omega_1, \omega_2, \ldots, \omega_N]^T$. Furthermore, the obtained sum is η.

Theorem 1: For any vertex $\boldsymbol{v} = [v_1, v_2, \ldots, v_N]^T$ of $\mathrm{CH}(\Phi_k)$, a weight vector $\boldsymbol{\omega} = [\omega_1, \omega_2, \ldots, \omega_N]^T$ can be found such that $\boldsymbol{v}$ is the WSRM solution for that weight vector.

For Theorem 1 to hold, we need the algorithm to examine the points in N ordered nested loops. When this condition is not true, however, the proof is still valid for all vertices $\boldsymbol{v}$ with nonzero elements. Fortunately, this does not affect the algorithms that we propose in the next section. In this condition a weaker form of Theorem 1 holds.

Weaker Form of Theorem 1: For any vertex $\boldsymbol{v} = [v_1, v_2, \ldots, v_N]^T$ of $\mathrm{CH}(\Phi_k)$ with $v_n > 0$, regardless of the order in which the algorithm examines points, a weight vector $\boldsymbol{\omega} = [\omega_1, \omega_2, \ldots, \omega_N]^T$ can be found such that $\boldsymbol{v}$ is the WSRM solution for that weight vector.

The iterative spectrum balancing (ISB) algorithm [12], [13] is an example where only the weaker form of Theorem 1 holds.

Theorem 2: Consider the Minkowski sum $C = A \oplus B$, where A and B are two convex polytopes in N-D space. If facet F_A of A is parallel to facet F_B of B, then C contains a facet F_C parallel to F_A and F_B which is obtained by $F_C = F_A \oplus F_B$.

Lemma 2 and Theorem 2 explain why it is not possible to get close to some points on the boundary of the RR using a WSRM approach in some scenarios. For these scenarios, many parallel facets exist on $\mathrm{CH}(\Phi_k)$ for different tones k; this is the result of one bit granularity of bit-loadings, similarity of adjacent tones, and nonconvex ABLV sets. According to Theorem 2, these parallel facets add together to make a potentially large facet on the boundary of the RR. According to Lemma 2, on the other hand, only vertex points of the facet can be reached using a WSRM approach. In some cases, all of the vertices can be far away from the target point, as for the two user scenario studied in Section II. In the next section, we propose a computationally efficient technique to reach almost any point on the boundary of the RR in a multi-user DSL environment.

IV. Iterative Facet Dividing Algorithm (IFDA)

In this section, we introduce our algorithm, which reaches the optimal point on the boundary of the RR along any arbitrary direction $\boldsymbol{R}_{\mathrm{target}}$. This is our main contribution in this paper, which is based on the investigations in the previous section.

As we showed earlier, when the optimal solution to the SB problem (7) is a nonvertex point of a facet, it cannot be reached using a WSRM approach. For $N = 2$, each facet is in fact a line in 2-D and has exactly two vertices on its boundary. However, for $N > 2$, a facet might have more than N vertices on its boundary. The convex hull of any selection of N vertices located at the boundary of a facet forms a subset of the facet. Any point $\boldsymbol{p}$ on the facet can be written as a weighted sum of at least one N-subset[1] of the vertices on its boundary, i.e., $\boldsymbol{p} = \sum_{n=1}^{N} \rho_n \boldsymbol{v}_n$. The weight factors, ρ_n, are positive and

[1]An N-set is defined as a set with N elements.

Authorized licensed use limited to: Katholieke Universiteit Leuven. Downloaded on June 22, 2009 at 05:18 from IEEE Xplore. Restrictions apply.

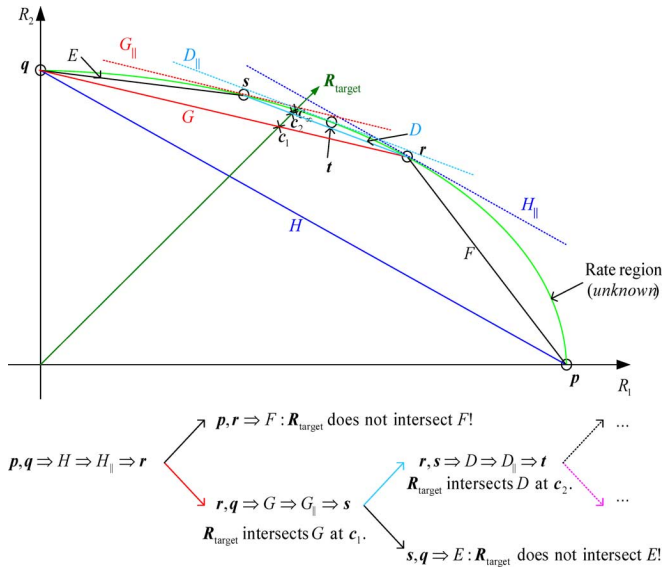

Fig. 2. An illustration of Algorithm 1.

sum to one (as in (13)). Obviously, to reach an arbitrary point on a facet, we need to find N vertices that include the point in their convex hull. Therefore, finding the optimal weight vector, $\boldsymbol{w} = [w_1, w_2, \ldots, w_N]^T$, is not enough for finding the optimal solution, because according to Lemma 2, the WSRM approach yields only one of the vertices.[2]

Three main algorithms have been proposed so far for finding the optimal weight vector $\boldsymbol{w}$, namely, bisection search using N nested loops [1], subgradient search [4], and subgradient search with adaptive step size [5]. Although, according to Theorem 1, for each vertex of the facet we can find a particular weight vector such that it is the WSRM solution for that weight vector, it is not clear how we can find N proper vertices that surround the target point using the three aforementioned algorithms. Algorithm 1 can find the required N vertices on the optimal facet in $\mathcal{O}((\log(K) + \log(b_{\max}))N/\log(N))$ iterations.

The iterative facet dividing algorithm (IFDA) starts with N basic vertices of the RR, called current facet indicator (CFI) points. It is essential that the facet limited to these points spans the whole search space, that is, any potential target bit-rate vector should intersect it. To achieve this goal, the first CFI points are calculated by setting the weight vector equal to the N standard basis vectors $\boldsymbol{e}_1$ to $\boldsymbol{e}_N$ in N-D. Then, in each iteration, a new vertex is found which replaces one of the vertices in CFI. The eliminated vertex is selected such that the new facet still covers the target solution. In fact, N subfacets can be calculated corresponding to N vertices of the current facet and $\boldsymbol{R}_{\text{target}}$ intersects at least one of them. When $\boldsymbol{R}_{\text{target}}$ intersects more than one subfacet, this means that it passes through the boundary of those subfacets. The subfacet (or one of the subfacets) intersected by $\boldsymbol{R}_{\text{target}}$ replaces the current facet and all other subfacets will be eliminated from the search space.

[2]Care should be taken in order not to confuse weight factors w_1 to w_N, which determine the normal vector to the facet, with the weight factors ρ_1 to ρ_N, which indicate the weight of the selected boundary vertices of a facet to reach a point on it. Although, we assume $\sum_{n=1}^{N} w_n^2 = 1$, weight factors w_1 to w_N can be scaled without loss of generality. However, ρ_1 to ρ_N cannot be scaled and we always have $\sum_{n=1}^{N} \rho_n = 1$.

The algorithm then iterates this process until the last calculated vertex is on the current facet. At this point, we have found N points on the boundary of the RR surrounding the optimal solution. Algorithm 2 uses these points to calculate the required bit-loadings and PSD values to reach the optimal solution.

In order to check whether $\boldsymbol{R}_{\text{target}}$ intersects a facet, we first calculate the point where $\boldsymbol{R}_{\text{target}}$ intersects the hyperplane H passing through the facet. Let $\boldsymbol{\omega}_H^T \boldsymbol{x} = \eta_H$ denote the equation of H. Using some algebra, the intersection point can be calculated by

$$\boldsymbol{p}_c = \frac{\eta_H}{\boldsymbol{\omega}_H^T \boldsymbol{R}_{\text{target}}} \boldsymbol{R}_{\text{target}}. \tag{16}$$

We then check if $\boldsymbol{p}_c$ can be written in the following form: $\boldsymbol{p}_c = \sum_{n=1}^{N} \rho_n \boldsymbol{p}_n$, where $\rho_n \geq 0$ and $\sum_{n=1}^{N} \rho_n = 1$. This can be easily done by calculating $\boldsymbol{\rho} = [\rho_1, \rho_2, \ldots, \rho_N]^T$ as follows:

$$\boldsymbol{\rho} = \left(\boldsymbol{P}_{\mathbf{xtd}}^T \boldsymbol{P}_{\mathbf{xtd}}\right)^{-1} \boldsymbol{P}_{\mathbf{xtd}}^T \begin{bmatrix} \boldsymbol{p}_c \\ 1 \end{bmatrix} \tag{17}$$

where $\boldsymbol{P}_{\mathbf{xtd}}$ is a $(N+1) \times N$ matrix defined by

$$\boldsymbol{P}_{\mathbf{xtd}} = \left[\begin{array}{c|c|c|c} \boldsymbol{p}_1 & \boldsymbol{p}_2 & \cdots & \boldsymbol{p}_N \\ 1 & 1 & & 1 \end{array}\right].$$

If all elements of $\boldsymbol{\rho}$ are greater than or equal to zero, then R_{target} intersects the facet. Otherwise, R_{target} does not intersect the facet.

Fig. 2 illustrates how the IFDA works for a simple two-user scenario. In Step 1, points $\boldsymbol{p}$ and $\boldsymbol{q}$ are calculated by setting the weight vector equal to basis unit vectors $\boldsymbol{e}_1 = [1, 0]^T$ and $\boldsymbol{e}_2 = [0, 1]^T$. The points $\boldsymbol{p}$ and $\boldsymbol{q}$ make the initial list of the CFI points. In Step 2, the line connecting $\boldsymbol{p}$ to $\boldsymbol{q}$ is calculated which plays the role of the hyperplane H. In Step 3, point $\boldsymbol{r}$, which is the WSRM solution with weight vector $\boldsymbol{\omega}_H$, is calculated. As it can be seen, $\boldsymbol{r}$ is the point of tangency of line $H_{||}$ with the rate region, where line $H_{||}$ is parallel to line H. As $\boldsymbol{r}$ is not on H, the algorithm continues to Step 5. In Step 5, we calculate $N = 2$ facets (here lines F and G) and verify which facet is intersected by $\boldsymbol{R}_{\text{target}}$. Lines G and F are obtained by replacing the first and second CFI points, $\boldsymbol{p}$ and $\boldsymbol{q}$, with the newly calculated point $\boldsymbol{r}$. As it can be seen, line G intersects $\boldsymbol{R}_{\text{target}}$ at point $\boldsymbol{c}_1$. Therefore, in Step 6 point $\boldsymbol{r}$ replaces $\boldsymbol{p}$ in the list of CFI points and we return to Step 2. The next point to be found is $\boldsymbol{s}$, which is the tangency point of $G_{||}$ with the RR. Then facet G is divided into facets D and E, where facet D intersects $\boldsymbol{R}_{\text{target}}$ at $\boldsymbol{c}_2$. The next calculated point is $\boldsymbol{t}$, which is the tangency point of $D_{||}$ with the RR. The algorithm continues until the calculated point in Step 3 is located on the facet calculated in Step 2. The final N CFI points are all located on the optimal facet of the RR. Moreover, it is guaranteed that the RD optimal point, $\boldsymbol{c}_\infty$, is in their convex hull.

A. Calculation of RD Optimal Point

Algorithm 1 gives us N points (or answers), say $\boldsymbol{p}_1$ to $\boldsymbol{p}_N$, surrounding the RD optimal point on the boundary of the RR. Let F_∞ and $\boldsymbol{c}_\infty$ denote the final facet and its intersection point with $\boldsymbol{R}_{\text{target}}$. It is guaranteed that $\boldsymbol{c}_\infty$ can be written as a weighted sum of $\boldsymbol{p}_1$ to $\boldsymbol{p}_N$. Assume we have stored the vectors of bit-loadings corresponding to point $\boldsymbol{p}_m (1 \leq m \leq N)$ in an $N \times K$ matrix $P_m = \left[\boldsymbol{b}_1^{(m)} | \boldsymbol{b}_2^{(m)} | \cdots | \boldsymbol{b}_K^{(m)}\right]$, where $\boldsymbol{b}_k^{(m)}$ is the

Authorized licensed use limited to: Katholieke Universiteit Leuven. Downloaded on June 22, 2009 at 05:18 from IEEE Xplore. Restrictions apply.

$N\times 1$ vector of bit-loading on tone k. The nth element of $\boldsymbol{p}_m$ is then equal to $\sum_{k=1}^{K}[P_m]_{n,k}$. Let $T=\left[\boldsymbol{b}_1^{(\mu_1)}|\boldsymbol{b}_2^{(\mu_2)}|\cdots|\boldsymbol{b}_K^{(\mu_K)}\right]$, where μ_1 to μ_K are integer numbers between 1 and N. In fact, T is a matrix of bit-loadings with each of its vectors chosen from one of the available vertices $\boldsymbol{p}_1$ to $\boldsymbol{p}_N$ on facet F_∞. Matrix T corresponds to point $\boldsymbol{t}$ with the nth element $t_n=\sum_{k=1}^{K}[T]_{n,k}$. From Lemma 2, $\boldsymbol{t}$ is located on facet F_∞. By changing the indices μ_1 to μ_K, we can reach different points on facet F_∞ located between points $\boldsymbol{p}_1$ to $\boldsymbol{p}_N$. The achievable points are at most $b_{\max}$ bits apart in each dimension, but in most cases almost any point is achievable. Our goal is to find a point as close as possible to point $\boldsymbol{c}_\infty$, where $\boldsymbol{R}_{\text{target}}$ intersects F_∞. The greedy Algorithm 2 can be used for this purpose.

Algorithm 1: The IFDA for finding N vertices surrounding the RD optimal point

Step 1) Find N current facet indicator (CFI) points $\boldsymbol{p}_1$ to $\boldsymbol{p}_N$ by setting the weight vector equal to the N standard basis vectors $\boldsymbol{e}_1$ to $\boldsymbol{e}_N$, where $\boldsymbol{e}_n \triangleq [\overbrace{0,\ldots,0}^{n-1\text{times}},\ 1,\ \overbrace{0,\ldots,0}^{N-n-1\text{times}}\ ]^T$.

Step 2) Calculate the hyperplane H passing through the CFI points by solving (15). Let $\boldsymbol{\omega}_H^T\boldsymbol{x}=\eta_H$ denote the equation of H.

Step 3) Find $\boldsymbol{r}$, the solution to a WSRM problem with weight vector $\boldsymbol{\omega}_H$.

Step 4) If $\boldsymbol{\omega}_H^T\boldsymbol{r}=\eta_H$ then exit.

Step 5) Calculate N subfacets each by replacing one of the CFI points with $\boldsymbol{r}$, and find the subfacet that the ray $\boldsymbol{R}_{\text{target}}$ (or its extension) intersects with.

Step 6) Set the CFI points equal to the vertices of the subfacet intersecting $\boldsymbol{R}_{\text{target}}$ and return to Step 2.

Algorithm 2: Finding the indices μ_1 to μ_N

Step 1) Set $\boldsymbol{t}\leftarrow\boldsymbol{p}_1$ and $\mu_k\leftarrow 1$ for $1\le k\le K$

Step 2) For all tones k from 1 to K {

Step 2.1) $\boldsymbol{t}\leftarrow\boldsymbol{t}-\boldsymbol{b}_k^{(\mu_k)}$;

Step 2.2) $\mu_k\leftarrow\arg\min_m \angle\left(\boldsymbol{t}+\boldsymbol{b}_k^{(m)},\boldsymbol{R}_{\text{target}}\right)$;

Step 2.3) $\boldsymbol{t}\leftarrow\boldsymbol{t}+\boldsymbol{b}_k^{(\mu_k)}$ };

Step 3) Repeat Step 2 until no improvements can be made, then terminate!

In Step 2.2 of Algorithm 2, $\angle(\boldsymbol{u},\boldsymbol{v})$ denotes the angle between vectors $\boldsymbol{u}$ and $\boldsymbol{v}$, which is a nonnegative number smaller than π and is calculated by $\angle(\boldsymbol{u},\boldsymbol{v})=\arccos\{\boldsymbol{u}^T\boldsymbol{v}/(|\boldsymbol{u}||\boldsymbol{v}|)\}$. As it can be seen, we first assume that point $\boldsymbol{p}_1$ is the optimal answer and set all of the indices μ_k to one. Then, for each of the available tones, the index is examined to see if we can get closer to the target by choosing the vector of bit-loading corresponding to another point on the final facet. Since by changing an index in a tone, better solutions might be found in other tones, we examine the index of all of the tones again, until no improvement can be made. At the end of the algorithm, the obtained bit-loading vectors are $\boldsymbol{b}_1^{(\mu_1)},\boldsymbol{b}_2^{(\mu_2)},\ldots,\boldsymbol{b}_K^{(\mu_K)}$, and the corresponding PSD values can be calculated using (3). It can be shown that the answer is at most $b_{\max}$ bits away from the optimal answer in each dimension. In most of our simulations, the results were at most only two bits away in each dimension.

B. Computational Complexity

Step 3 of the IFDA involves exhaustive searches on the tones, which has a computational complexity of $\mathcal{O}(K(b_{\max}+1)^N)$. Therefore, the number of iterations of Step 3 determines the order of the computational complexity of the whole algorithm. Thus, we only discuss the number of iterations of Algorithm 1. Calculation of the first N points corresponding to standard basis vectors $\boldsymbol{e}_1$ to $\boldsymbol{e}_N$ does not require carrying out an exhaustive search. The point corresponding to $\boldsymbol{e}_n$ can be simply calculated by setting the spectrum of user n at the PSD mask level and the spectrum of other users to zero. In each iteration, the rate region is partitioned into N subsets, and all subsets but one will be eliminated from the search space. That is, on the average, the search space is divided by N in each iteration. The number of points on the boundary of the RR is upper-bounded by $N(Kb_{\max})^{N-1}$ which is half the number of points on the boundary of an N-D cube with edge length $Kb_{\max}$. Therefore, for the worst-case scenario, where the final facet has only one point on it, the number of iterations i can be calculated by $N(K(b_{\max}+1))^{N-1}/N^{\mathrm{i}}=1$. Thus, we get $i=1+(N-1)\log(K(b_{\max}+1))/\log(N)$, which means the number of iterations is $\mathcal{O}((\log(K)+\log(b_{\max}))\times N/\log(N))$. For example, for 2 and 7 VDSL users using 997 bandplan at US direction with $K=1635$ and $b_{\max}=15$, we get $i\approx 16$ and 32, respectively. In practice, the number of iterations can be much smaller. For example, by running simulations for 90 uniformly distributed points on the RR of the 2-user US VDSL scenario of Section II, the average number of iterations is 3.22.

Comparison of the proposed algorithm to the bisection search [1], the subgradient search [4], and the step-adaptive subgradient search [5] is not sensible as those algorithms cannot be used to find the N points surrounding the optimal point on the target facet. Here we briefly review the number of iterations needed for those algorithms to find the closest point to the optimal point. For the bisection search, the number of iterations is $\mathcal{O}((\log(Kb_{\max}))^N)$, which is exponential in the number of users. The number of iterations for the subgradient search and the step-adaptive subgradient search has not been evaluated analytically in the literature. Reference [5] suggests that the step-adaptive subgradient search is much faster than the subgradient search. The number of iterations for the step-adaptive subgradient search for ADSL scenarios with 2 to 4 users has been reported to be 100 to 150; this number drops to less than 40 with previous knowledge of the step size. Obviously, the proposed algorithm is much faster than the subgradient search family of algorithms as well. There are two main reasons for the faster convergence of the proposed algorithm. Firstly, it updates the weight vector based on N previous points on the rate region, while the subgradient search family updates the weight vector on the last point only. Secondly, the distance to the optimal point is monotonically decreasing for the algorithm, while the subgradient search family fluctuates around the optimal point a few

Authorized licensed use limited to: Katholieke Universiteit Leuven. Downloaded on June 22, 2009 at 05:18 from IEEE Xplore. Restrictions apply.

times particularly as they get close to the optimal point. Furthermore, subgradient algorithms cannot easily handle the cases where the final facet is large and might need many iterations to converge. By contrast, large facets reduce the number of iterations needed for our algorithm.

C. Extension of the Results

We proposed the algorithms in this paper to overcome difficulties in OSB for MLAP DSL systems. However, the proposed IFDA can be used for finding the optimal weight factors for DSL systems with aggregate power constraints as well. The algorithm can be used in many other similar weighted sum optimization problems. Some examples in the same context include OSB in the general case [1], suboptimal SB schemes for DSL [10]–[16], partial crosstalk cancellation for DSL [3], and joint partial crosstalk cancellation and spectrum management [17]–[19]. Whenever needed, Lagrange multipliers should be separately found for each set of weight factors, e.g., in an inner loop. To do this we may use the bisection or subgradient search. An interesting problem for investigation is if the IFDA can be used to find the Lagrange multipliers as well, or if we can modify the IFDA to find them jointly with the weight factors.

It is important to note that Algorithm 1 (IFDA) can always be used for the aforementioned problems, even when the RR is not convex. However, in some cases, we might not be allowed to use Algorithm 2. For example, consider OSB with tight AFE power constraints. When the RR is not strictly convex, the IFDA provides N points on the CH of the RR. The AFE power constraints have been met for all of the modems in all of the N solutions. However, application of Algorithm 2 can violate power constraint for some of the modems. This is indeed inevitable under the assumption of nonconvexity of the RR. In these cases, we can use time sharing between the solutions to reach the desired point on the boundary of the RR. The duration of time devoted to solution n should be proportional to the weight factor ρ_n.

In many cases of the general OSB problem, such as very short or very long DSL loops, we cannot predict if the optimal PSD is restricted by the AFE power constraints prior to solving the OSB problem. In these situations, Lagrange multipliers go to zero when the AFE power constraint is not binding. Again WSRM approach may fail and thus cannot be trusted. This is a reason why we believe our proposed technique should be used in any scenario, even for modems with tight AFE power constraints.

V. Simulation Results

In this section, we review a few simulation results. All simulations have been carried out on VDSL loops with the simulation parameters listed in Table I. For the first simulation, we tested our proposed algorithms for the 2 user scenario of Section II-B. For this scenario, the final facet is a line segment in 2-D and the two points at its ends are $\boldsymbol{p}_1 = [24501, 1513]^T$ and $\boldsymbol{p}_2 = [1513, 24501]^T$. Fig. 3 illustrates the bit-loadings corresponding to these points. As it can be seen, the bit-loadings corresponding to $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ are reversed; this is because the loops are completely similar and the scenario is symmetrical.The information in Fig. 3 is the input to Algorithm 2. The point calculated by Algorithm 2 is $[13007, 13007]^T$, which is the optimal achievable point maximizing $R_1 = R_2$. Multiplying the answer

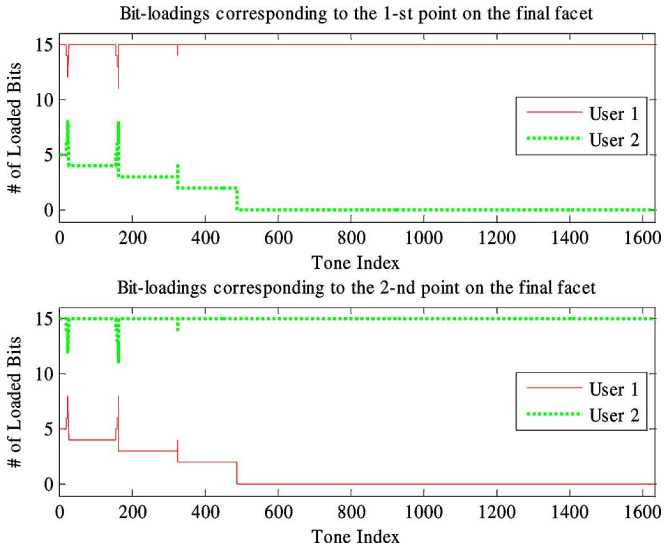

Fig. 3. Bit-loadings corresponding to the points on the final facet, for the 2-user VDSL scenario of Section II-B.

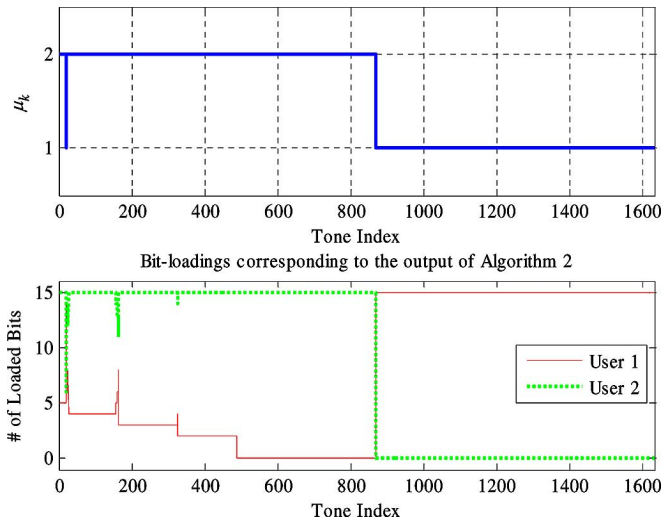

Fig. 4. Output of Algorithm 2, including the indices μ_k and the corresponding bit-loadings for the 2-user VDSL scenario of Section II-B.

by the DMT symbol rate we get $r_1 = r_2 = 52.028\,\text{Mbps}$, which is 12.6% higher than that of SSM ($r_1 = r_2 = 46.20\,\text{Mbps}$) for both users, and 759.6% higher than the bit-rate of at least one of the users for the result of the WSRM approach. Fig. 4 shows the output of Algorithm 2, including the indices μ_1 to μ_K and the corresponding bit-loadings for each user. As it can be seen, for almost half of the tones, the indices are two and the rest are one.

We have simulated the same scenario for three to five users, as well. Fig. 5 shows the number of iterations versus N for Algorithm 1. Clearly, the number of iterations is even smaller than the estimated value in Section IV. This is firstly because the number of points on the RR is smaller than the assumed worst-case value, and secondly, because the number of points on the final facet is much larger than one.

Simulation results for $N = 5$ are presented in Fig. 6. The five points on the final facet are calculated by the algorithm as $\boldsymbol{p}_1 = [9317, 8354, 7671, 8723, 10474]^T$, $\boldsymbol{p}_2 =$

Authorized licensed use limited to: Katholieke Universiteit Leuven. Downloaded on June 22, 2009 at 05:18 from IEEE Xplore. Restrictions apply.

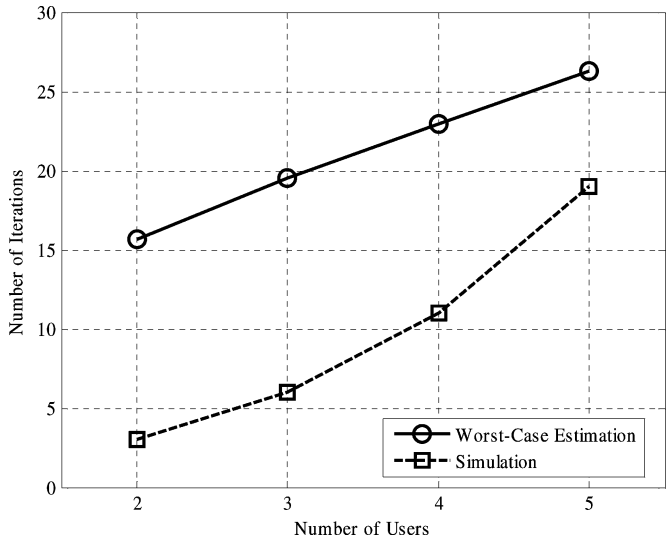

Fig. 5. Number of iterations versus number of users for Algorithm 1.

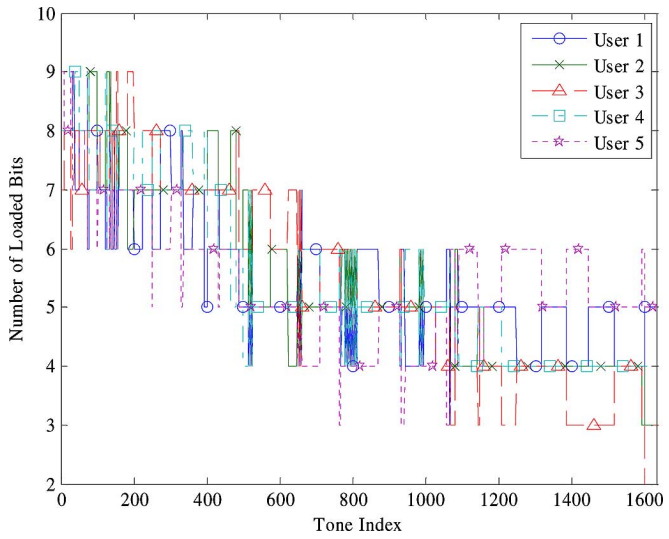

Fig. 6. Bit-loadings for a 5-user 250 m cable US VDSL scenario.

$[7407, 8981, 9470, 10210, 8471]^T$, $\boldsymbol{p}_3 = [9320, 7674, 10474, 8351, 8720]^T$, $\boldsymbol{p}_4 = [8720, 10474, 9320, 7674, 8351]^T$, and $\boldsymbol{p}_5 = [10220, 8974, 8244, 9574, 7527]^T$. Once again, the scenario is symmetric and as it can be seen, points $\boldsymbol{p}_3$ and $\boldsymbol{p}_4$ are permutations of each other. However, not all of the points are permutations of one another; this shows that the final facet includes more than 5 vertices (at least $4 \times 5! = 480$). In this scenario, however, the algorithm has smartly selected five points on the final facet that surround the target point. The calculated point of Algorithm 2 for this scenario is $\boldsymbol{t} = [8908, 8907, 8908, 8908, 8908]^T$. Taking into account the one-bit granularity of the bit-loading, the calculated point is an optimum solution for the problem. Note that any of the points $\boldsymbol{p}_1$ to $\boldsymbol{p}_5$ are acceptable solutions to the problem using the WSRM approach. However, for any of them, at least one of the users receives less than 85% of the achievable bit-rate. Obviously, only our technique is capable of sharing the resources fairly among the users.

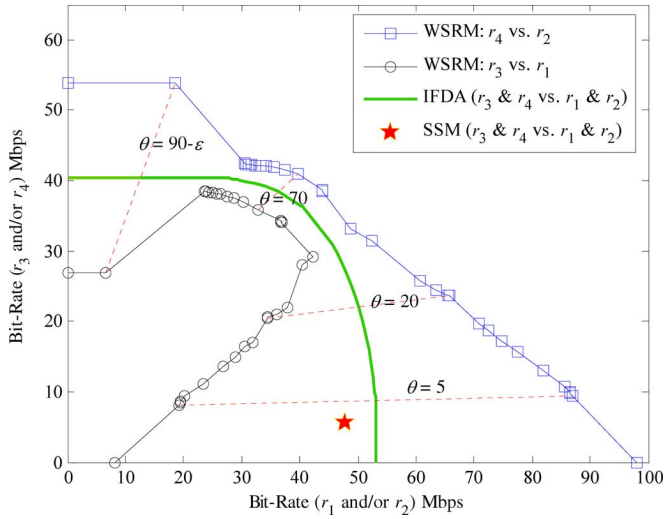

Fig. 7. Simulation results for a 4-user near-far US VDSL scenario. Users 1 and 2 are located at 200 m and users 3 and 4 are located at 500 m away from the line-termination node. A few points corresponding to the same value of θ are connected to each other using dashed ines. ε is a sufficiently small positive number.

Comparing the optimal answer for the 2-user scenario with the 5-user scenario, we see that the optimal bit-loadings for the 2-user scenario are obtained by a frequency division multiplexing approach. That is, for most of the tones, one user transmits the maximum number of bits and the other user is turned off. Whilst for the 5-user scenario all of the users transmit at all of the tones simultaneously. This implies that the bit-loading sets, Φ_k, are more likely to be convex (or closer to being convex) for greater number of users. This reduces the size of parallel facets in different tones and the size of the final facet on the RR. The upshot of which is that more iterations are needed for convergence. However, as it can be seen in Fig. 5, the number of iterations is still smaller than the estimated worst-case value.

Fig. 7 illustrates the simulation results for a four-user near-far US VDSL scenario. For this scenario, users 1 and 2 are located at 200 m and users 3 and 4 are located at 500 m away from the line-termination node. All users share the first 200 m of the cable, but only loops 3 and 4 are adjacent to each other on the last 300 m of the cable. We have simulated OSB using WSRM approach with equal weight factors for users 1 and 2, and users 3 and 4. That is $\boldsymbol{w} = [\cos(\theta), \cos(\theta), \sin(\theta), \sin(\theta)]^T/\sqrt{2}$, where $\theta \in [0, 90]$ degrees. Our algorithm, on the other hand, is simulated for $\boldsymbol{R}_{\text{target}} = [\cos(\varphi), \cos(\varphi), \sin(\varphi), \sin(\varphi)]^T$, where again $\varphi \in [0, 90]$ degrees. The horizontal coordinate is used for r_1 and r_2 and the vertical coordinate is used for r_3 and r_4. Each value of θ or φ gives us a vector of bit-rates $[r_1, r_2, r_3, r_4]^T$. Since the target bit-rates for users 1 and 2 are set to be equal and the target bit-rates for users 1 and 2 are set to be equal, we get $r_1 \approx r_2$ and $r_3 \approx r_4$ for our algorithm. However, WSRM approach is not capable of providing the same bit-rates for users one and two, and users three and four. Therefore, we have plotted r_3 versus r_1 and r_4 versus r_2. In order to give an idea of the relation of points on the two curves, a few points corresponding to the same value of θ are connected to each other using dashed lines. As it can be seen, there exists a huge asymmetry in the bit-rates of the similar pairs using the WSRM ap-

Authorized licensed use limited to: Katholieke Universiteit Leuven. Downloaded on June 22, 2009 at 05:18 from IEEE Xplore. Restrictions apply.

proach. That is users 2 and 4 achieve bit-rates much higher than users 1 and 3, respectively.

VI. Conclusion

The WSRM approach always provides a point on the boundary of the RR for MLAP DSL systems, even if the RR is not convex. The point will always be on the CH of the RR. Although any point on the CH of the RR is optimal for at least one vector of weight factors, the WSRM approach is not capable of reaching some of these points. These points are located on facets of the CH. Sometimes the size of these facets is very large which results in a poor performance for OSB based on the WSRM approach. The IFDA algorithm proposed in this article is capable of efficiently and reliably finding the proper number of vertices surrounding the target point on the corresponding facet. The bit-loadings and PSD values of the vertices can then be exploited to approximate the target point very closely using the second algorithm proposed in this paper.

Appendix
Properties of Convex Hull of ABLV Sets

This appendix lists the lemmas and theorems introduced in Section III with their proofs.

Lemma 1: A vertex of $\mathrm{CH}(\Phi_k)$ is a member of Φ_k.

Proof: This lemma follows immediately from (13). Clearly, a vertex point $\boldsymbol{v}$ cannot be written in the $\boldsymbol{v} = \sum_{n=1}^{N} \omega_n \boldsymbol{p}_n$ form using other members of Φ_k. This can only happen when $\boldsymbol{v} = \boldsymbol{p}_m$ for an arbitrary index $m (1 \leq m \leq N)$, $\omega_n = 0$ for $n \neq m$, and $\omega_m = 1$. ■

Lemma 2: The solution to the WSRM problem for tone k, i.e., $\left(b_k^1, \ldots, b_k^N\right) = \arg\max \boldsymbol{\mathcal{W}}_k$, is a vertex of $\mathrm{CH}(\Phi_k)$.

Proof: The equation $\sum_{n=1}^{N} w_n b_k^n = \max \sum_{n=1}^{N} w_n b_k^n$ indicates a hyperplane H in the N-D space which divides the space into two subspaces with $\sum_{n=1}^{N} w_n b_k^n \leq \max \sum_{n=1}^{N} w_n b_k^n$ and $\sum_{n=1}^{N} w_n b_k^n \geq \max \sum_{n=1}^{N} w_n b_k^n$. Therefore, the hyperplane H is always tangent to $\mathrm{CH}(\Phi_k)$. Two scenarios may occur when maximizing $\boldsymbol{\mathcal{W}}_k$ as illustrated for $N = 2$ in Fig. 8(a) and (b). It is clear that the lemma holds, when only one vertex point $\boldsymbol{p}$ maximizes $\boldsymbol{\mathcal{W}}_k$ as shown in Fig. 8(a). As a vertex point $\boldsymbol{p}$ is always a member of Φ_k, $\boldsymbol{p}$ is selected by the algorithm without any ambiguity. The second scenario happens when more than one point maximizes $\boldsymbol{\mathcal{W}}_k$, as shown in Fig. 8(b). In these cases, we can see that the area containing the optimal points are always bounded by vertices of the convex hull, here points $\boldsymbol{p}$ and $\boldsymbol{q}$. Note that all of the points on the line connecting $\boldsymbol{p}$ and $\boldsymbol{q}$ (including $\boldsymbol{p}$ and $\boldsymbol{q}$) maximize the WSB. Since Φ_k is not convex, some of the points are not members of Φ_k. However, from Lemma 1, $\boldsymbol{p}$ and $\boldsymbol{q}$ are always members of Φ_k. When multiple points maximize $\boldsymbol{\mathcal{W}}_k$, the algorithm does not strictly specify which point should be selected. This depends on the implementation of the algorithm. The algorithm tests all of the points one-by-one to find out which one maximizes $\boldsymbol{\mathcal{W}}_k$. A natural implementation, which we assume here, is to calculate $\boldsymbol{\mathcal{W}}_k$ in the most inner loop of N nested loops, where in the nth inner loop, b_n ranges from 0 to $b_{\max}$. The best vector of bit-loading and the corresponding WSB will be stored in temporary variables and are changed only if we find a member of Φ_k with a greater WSB. Since all of the points on the line connecting $\boldsymbol{p}$ and $\boldsymbol{q}$ have the same WSB value, the algorithm simply takes the first examined point on this line. The point that is examined first is the point with the smallest b_1 value. For the example of Fig. 8(b), point $\boldsymbol{q}$ will be chosen by the algorithm as it has the minimum b_1 value among all points that maximize the WSB. If there is more than one point with maximum WSB and minimum b_1 value, the point with the smallest b_2 is chosen. Clearly, the selected point cannot be located between two points on the boundary of the convex hull. This rule can be generalized to N dimensions. Taking into account Lemma 1, this means that the selected point is always a vertex of the convex hall of Φ_k. ■

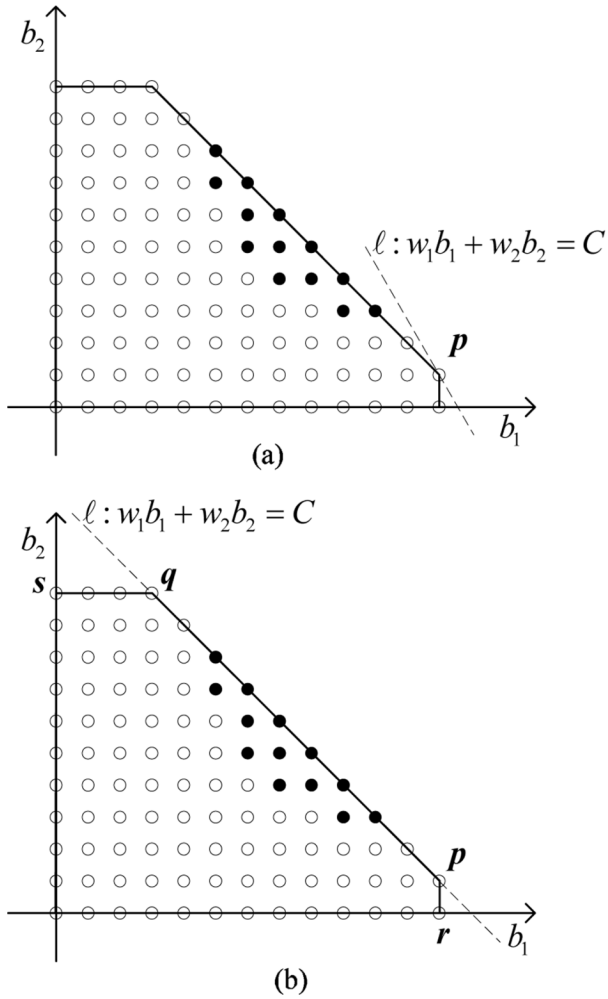

Fig. 8. Possible solutions to a WSRM problem for an arbitrary tone k. The white and black circles indicate achievable and unachievable points, respectively, and the boundary of $\mathrm{CH}(\Phi_k)$ is indicated by solid lines. The dashed line ℓ is the subspace indicating $w_1 b_1 + w_2 b_2 = C$ where $C = \max(w_1 b_1 + w_2 b_2)$ and $b = [b_1, b_2]^T \in \mathrm{CH}(\Phi_k)$. (a) A single point maximizes the WSB. (b) More than one point maximize the WSB.

Lemma 3: If (14) describes a hyperplane passing through a facet F on $\mathrm{CH}(\Phi_k)$, then the points belonging to Φ_k on F maximize a WSB with weight vector $[w_1, w_2, \ldots, w_N]^T = [\omega_1, \omega_2, \ldots, \omega_N]^T$. Furthermore, the obtained sum is η.

Proof: Since a hyperplane divides the N-D space into two subspaces with $\sum_{n=1}^{N} w_n b_k^n \leq \max \sum_{n=1}^{N} w_n b_k^n$ and

Authorized licensed use limited to: Katholieke Universiteit Leuven. Downloaded on June 22, 2009 at 05:18 from IEEE Xplore. Restrictions apply.

$\sum_{n=1}^{N} w_n b_k^n \geq \max \sum_{n=1}^{N} w_n b_k^n$, the result follows immediately. ∎

Theorem 1: For any vertex $\boldsymbol{v} = [v_1, v_2, \ldots, v_N]^T$ of $\mathrm{CH}(\Phi_k)$, a weight vector $\boldsymbol{\omega} = [\omega_1, \omega_2, \ldots, \omega_N]^T$ can be found such that $\boldsymbol{v}$ is the WSRM solution for that weight vector.

Proof: We prove this theorem in two steps. In the first step, we assume that all elements of $\boldsymbol{v}$ are greater than zero, i.e., $v_n > 0$ for $1 \leq n \leq N$. Next, we assume $\boldsymbol{v}$ is the intersection of M facets F_1 to F_M of $\mathrm{CH}(\Phi_k)$ with equations

$$\sum_{n=1}^{N} \omega_n^{(m)} b_n = \eta^{(m)} \tag{18}$$

where, without loss of generality, we assume that the constants $\eta^{(m)}$ are positive. Using this assumption, here we prove that $\omega_n^{(m)} \geq 0$ for all m and n. Consider the point $\boldsymbol{v}^{(n)} = [v_1, \ldots, v_{n-1}, 0, v_{n+1}, \ldots, v_N]^T$, which is obtained by zeroing the nth element of $\boldsymbol{v}$. Since $\boldsymbol{v} \in \Phi_k$, then $\boldsymbol{v}^{(n)} \in \Phi_k$. For $\boldsymbol{v}$, we have $\sum_{\ell=1}^{N} \omega_\ell^{(m)} v_\ell = \eta^{(m)}$. For $\boldsymbol{v}^{(n)}$, we have $\sum_{\substack{\ell=1;\\ \ell \neq n}}^{N} \omega_\ell^{(m)} v_\ell \leq \eta^{(m)}$. Subtracting the inequality from the equation, we get $\omega_n^{(m)} v_n \geq 0$. Finally, since $v_n > 0$, we have $\omega_n^{(m)} \geq 0$.

Now consider M arbitrary coefficients $\kappa_m (\kappa_m > 0)$ for $m = 1$ to M. Construct coefficients σ_n for $n = 1$ to N as follows:

$$\sigma_n = \sum_{m=1}^{M} \kappa_m \omega_n^{(m)} \tag{19}$$

which are all nonnegative since $\omega_n^{(m)} \geq 0$. We assume κ_m are normalized such that $\sum_{n=1}^{N} \sigma_n^2 = 1$. Then, $\boldsymbol{v}$ is the solution to the WSRM problem with coefficients σ_n. To prove this, we note that (18) implies that $\sum_{n=1}^{N} \sigma_n v_n = \sum_{m=1}^{M} \kappa_m \eta^{(m)}$. However, for any other point $\boldsymbol{u} \in \Phi_k$, $\sum_{n=1}^{N} \omega_n^{(m)} b_n < \eta^{(m)}$ for at least one facet F_m and therefore $\sum_{n=1}^{N} \sigma_n u_n < \sum_{n=1}^{N} \sigma_n v_n$.

In the second step, we prove the theorem for the cases where $\boldsymbol{v}$ includes $L < N$ zeros at indices $z_1, z_2, \ldots, z_L$. Consider the subspace S_Z obtained by eliminating coordinates $z_1, z_2, \ldots, z_L$. We can use the proof in step 1 to show that $\boldsymbol{v}$ is a solution to a WSRM problem in S_Z. Therefore, there exists a vector $\boldsymbol{\omega}$ (with zeros at indices z_1 to z_L) such that $\sum_{n=1}^{N} \omega_n v_n$ is greater than $\sum_{n=1}^{N} \omega_n u_n$ for any $\boldsymbol{u} \in S_Z$. Now, returning to the N-D space, we consider an arbitrary point $\boldsymbol{t} \in \Phi_k$. We let $\boldsymbol{s}$ be the projection of $\boldsymbol{t}$ on S_Z; this is obtained by setting the elements z_1 to z_L of $\boldsymbol{t}$ to zero. Then, $\boldsymbol{s}$ is a member of Φ_k as well and it is examined by the algorithm sooner than $\boldsymbol{t}$. Therefore, we have $\sum_{n=1}^{N} \omega_n v_n > \sum_{n=1}^{N} \omega_n s_n$, or $\sum_{n=1}^{N} \omega_n v_n = \sum_{n=1}^{N} \omega_n s_n$ and $\boldsymbol{v}$ is examined by the algorithm sooner than $\boldsymbol{s}$. On the other hand, since $\boldsymbol{\omega}$ is zero at indices z_1 to z_L, we have $\sum_{n=1}^{N} \omega_n s_n = \sum_{n=1}^{N} \omega_n t_n$. Therefore, we either have $\sum_{n=1}^{N} \omega_n v_n > \sum_{n=1}^{N} \omega_n t_n$, or $\sum_{n=1}^{N} \omega_n v_n = \sum_{n=1}^{N} \omega_n t_n$ and $\boldsymbol{v}$ is examined by the algorithm sooner than $\boldsymbol{t}$, which completes the proof. ∎

Weaker Form of Theorem 1: For any vertex $\boldsymbol{v} = [v_1, v_2, \ldots, v_N]^T$ of $\mathrm{CH}(\Phi_k)$ with $v_n > 0$, regardless of the order in which the algorithm examines points, a weight vector $\boldsymbol{\omega} = [\omega_1, \omega_2, \ldots, \omega_N]^T$ can be found such that $\boldsymbol{v}$ is the WSRM solution for that weight vector.

Proof: Since we assume that the elements of $\boldsymbol{v}$ are nonzero, the first step of the proof for Theorem 1 holds here. ∎

Theorem 2: Consider the Minkowski sum $C = A \oplus B$, where A and B are two convex polytopes in N-D space. If facet F_A of A is parallel to facet F_B of B, then C contains a facet F_C parallel to F_A and F_B which is obtained by $F_C = F_A \oplus F_B$.

Proof: Since $F_A \| F_B$, a vector $\boldsymbol{\omega}$ exists such that for any $\boldsymbol{a}_0 \in F_A$ and $\boldsymbol{b}_0 \in F_B$, $\boldsymbol{\omega}^T \boldsymbol{a}_0 = \eta_A$ and $\boldsymbol{\omega}^T \boldsymbol{b}_0 = \eta_B$. Therefore, for any point $\boldsymbol{c}_0 = \boldsymbol{a}_0 + \boldsymbol{b}_0 \in F_C$ we have $\boldsymbol{\omega}^T \boldsymbol{c}_0 = \eta_A + \eta_B$. To prove that F_C is a facet, we need to show that F_C is located on the boundary of C. Since F_A and F_B are facets of A and B, for any point $\boldsymbol{c} = \boldsymbol{a} + \boldsymbol{b} \in C$ where $\boldsymbol{a} \in A$ and $\boldsymbol{b} \in B$, we have $\boldsymbol{\omega}^T (\boldsymbol{a} + \boldsymbol{b}) \leq \eta_A + \eta_B$. Therefore, F_C is a facet of C parallel to F_A and F_B. ∎

ACKNOWLEDGMENT

The author would like to thank D. R. Baqaee for his invaluable comments and suggestions.

REFERENCES

[1] R. Cendrillon *et al.*, "Optimal multiuser spectrum balancing for digital subscriber lines," *IEEE Trans. Commun.*, vol. 54, pp. 922–933, 2006.

[2] D. Statovci, T. Nordstrom, and R. Nilsson, "Spectrum balancing for DSL with restrictions on maximum transmit PSD," in *Proc. 2nd Int. Conf. Access Netw. (AccessNets'07)*, Ottawa, Canada, Aug. 2007, pp. 1–5.

[3] A. R. Forouzan and L. M. Garth, "Computationally efficient partial crosstalk cancellation in fast time-varying DSL crosstalk environments," *EURASIP J. Adv. Signal Process.*, vol. 2007, no. Article ID 72041, p. 15, 2007.

[4] W. Yu, R. Lui, and R. Cendrillon, "Dual optimization methods for multiuser orthogonal frequency division multiplex systems," in *Proc. IEEE Global Telecom. Conf. (GLOBECOM'04)*, Dallas, TX, Dec. 2004, pp. 225–229.

[5] P. Tsiaflakis *et al.*, "An efficient search algorithm for the Lagrange multipliers of optimal spectrum balancing in multi-user xDSL systems," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP'06)*, Toulouse, France, May 2006, vol. 4, pp. 101–104.

[6] E. W. Weisstein, Convex Hull MathWorld–A Wolfram Web Resource [Online]. Available: http://mathworld.wolfram.com/ConvexHull.html

[7] C. B. Barber, D. P. Dobkin, and H. T. Huhdanpaa, "The Quickhull algorithm for convex hulls," *ACM Trans. Math. Softw.*, vol. 22, no. 4, pp. 469–483, Dec. 1996.

[8] *Spectrum Management for Loop Transmission Systems*, ANSI Std. T1.417, 2003.

[9] *Very High Speed Digital Subscriber Line (VDSL); Functional Requirements*, ETSI Std. TS 101 270-1, V.1.3.1, 2003.

[10] R. Cendrillon *et al.*, "Autonomous spectrum balancing for digital subscriber lines," *IEEE Trans. Signal Process.*, vol. 55, no. 8, pp. 4241–4257, Aug. 2007.

[11] Y. Xu, T. Le-Ngoc, and S. Panigrahi, "Global concave minimization for optimal spectrum balancing in multi-user DSL networks," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 287–2885, Jul. 2008.

[12] R. Cendrillon and M. Moonen, "Iterative spectrum balancing for digital subscriber lines," in *Proc. IEEE Int. Conf. Commun. (ICC'05)*, Seoul, South Korea, May 2005, vol. 3, pp. 1937–1941.

[13] R. Lui and W. Yu, "Low-complexity near-optimal spectrum balancing for digital subscriber lines," in *Proc. IEEE Intl. Conf. Commun. (ICC'05)*, Seoul, South Korea, May 2005, vol. 3, pp. 1947–1951.

[14] D. D. Yu, K. Seong, and J. M. Cioffi, "Multiuser discrete bit-loading for digital subscriber lines," in *Proc. IEEE Int. Conf. Commun. (ICC'07)*, Glasgow, Scotland, Jun. 2007, pp. 2755–2760.

[15] J. Papandriopoulos and J. S. Evans, "Low-complexity distributed algorithms for spectrum balancing in multi-user DSL networks," in *Proc. IEEE Int. Conf. Commun. (ICC'06)*, Istanbul, Turkey, Jun. 2006, vol. 7, pp. 3270–3275.

[16] P. Tsiaflakis *et al.*, "Convex relaxation based low-complexity optimal spectrum balancing for multi-user DSL," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP'07)*, Honolulu, HI, Apr. 2007, vol. 3, pp. 349–352.

Authorized licensed use limited to: Katholieke Universiteit Leuven. Downloaded on June 22, 2009 at 05:18 from IEEE Xplore. Restrictions apply.

[17] J. Vangorp *et al.*, "Joint spectrum management and constrained partial crosstalk cancellation in a multi-user xDSL environment," *Signal Process.*, vol. 87, no. 12, pp. 3131–3146, 2007.

[18] C.-Y. Chen *et al.*, "Optimized resource allocation for upstream vectored DSL systems with zero-forcing generalized decision feedback equalizer," *IEEE J. Sel. Topics Signal Process.*, vol. 1, no. 4, pp. 686–699, Dec. 2007.

[19] V. M. K. Chan and W. Yu, "Joint multiuser detection and optimal spectrum balancing for digital subscriber lines," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Philadelphia, PA, Mar. 2005, vol. 3, pp. 333–336.



Amir R. Forouzan (S'99–M'04) received the B.S. and M.Sc. degrees in electrical engineering from Sharif University of Technology, Tehran, Iran, in 1998 and 2000, respectively, and the Ph.D. degree with highest distinction from University of Tehran in 2004.

From August 1999 to May 2004, he was with the Iran Telecommunication Research Center as a Research Fellow. From June 2004 to October 2008, he was with the University of Canterbury, Christchurch, New Zealand. Since November 2008, he has been with the Electrical Engineering Department, Katholieke Universiteit Leuven, Belgium. His research interests include dynamic spectrum management in DSL, MIMO and OFDM communication systems, ultrawideband radio, and wireless and optical CDMA.

Authorized licensed use limited to: Katholieke Universiteit Leuven. Downloaded on June 22, 2009 at 05:18 from IEEE Xplore. Restrictions apply.