This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2011 proceedings

Lagrange Multiplier Optimization for Optimal Spectrum Balancing of DSL with Logarithmic Complexity

Amir R. Forouzan and Marc Moonen
Dept. of Electrical Engineering (ESAT-SISTA), Katholieke Universiteit Leuven, Leuven, 3001, Belgium
Email: {amir.forouzan,marc.moonen}@esat.kuleuven.be

***Abstract*—Lagrange dual optimization (LDO) technique is a powerful tool for solving constrained optimization problems in and is generally considered to be optimal in the literature. LDO relaxes a constrained problem into an unconstrained dual problem using Lagrange multipliers. To solve the dual problem, the optimal value of the Lagrange multipliers should be found. The Lagrange multipliers are usually determined in an iterative process and reducing the number of iterations is of crucial importance to obtain systems with manageable computational complexity. In this paper, we show that for the LDO to be optimal in optimal spectrum balancing of DSL, the joint rate and power region (JRPR) should be strictly convex. Moreover, we propose a new LDO based algorithm with two advantages. Firstly, the computational complexity of the algorithm is logarithmic in the desired precision. Secondly, the algorithm can be used to find the optimal solution even when the JRPR is not strictly convex.**

***Index Terms*—Convex optimization, digital subscriber line (DSL), dual decomposition, dynamic spectrum management (DSM), non-convex optimization, resource allocation.**

I. Introduction

Lagrange dual optimization (LDO) techniques have attracted a lot of attention for solving constrained optimization problems in various fields of communications [1]–[6]. The most famous problem in this category is the optimal spectrum balancing (OSB) of DSL [1][1]. The OSB problem is stated as follows:

$$\text{maximize } R_1 \tag{1a}$$

$$\text{subject to } R_n \geq R_{\min}^{(n)} \text{ for all } n; 2 \leq n \leq N, \text{ and} \tag{1b}$$

$$P_n \leq P_{\max}^{(n)} \text{ for all } n; 1 \leq n \leq N, \tag{1c}$$

[1]In this paper, we concentrate on the OSB problem, however, our results can be generalized to a wide range of separable optimization problems in MIMO OFDM systems [2], communications in fading channels with quantized states [3], cognitive radios [4], joint routing and resource allocation [5], power allocation in the vector broadcast channels [6], etc.

This research work was carried out at the ESAT Laboratory of Katholieke Universiteit Leuven, in the frame of

- K.U.Leuven Research Council CoE EF/05/006 Optimization in Engineering (OPTEC),
- Concerted Research Action GOA-MaNet,
- The Belgian Programme on Interuniversity Attraction Poles initiated by the Belgian Federal Science Policy Office IUAP P6/04 'Dynamical systems, control and optimization' (DYSCO) 2007-2011,
- Research Project IBBT,
- Research Project FWO nr.G.0235.07 'Design and evaluation of DSL systems with common mode signal exploitation', and
- IWT Project 'PHysical layer and Access Node TEchnology Revolutions: enabling the next generation broadband network' (PHANTER).

The scientific responsibility is assumed by its authors.

where R_n and P_n are the bit-rate and aggregate transmit power of user n, $R_{\min}^{(n)}$ and $P_{\max}^{(n)}$ are the minimum required bit-rate and the maximum aggregate transmit power for user n, and N is the total number of users. The rate region (RR) is defined as the set of all achievable N-dimensional (N-D) vectors $(R_1, \ldots, R_N)^{\mathrm{T}}$ which satisfy (1c). To solve (1) using LDO, we maximize the following Lagrangian [1]:

$$\mathcal{L} \equiv \sum_{n=1}^{N} w_n R_n - \sum_{n=1}^{N} \lambda_n P_n \tag{2}$$

where $w_n \geq 0$ and $\lambda_n \geq 0$ are called the weight factor and Lagrange multiplier for user n. The solution is found by an iterative process in which w_n and λ_n are updated until all of the constraints in (1b) and (1c) are satisfied. The number of iterations plays an important role in the computational complexity of the technique and several techniques have been proposed to reduce the number of iterations including the bisection search [1], sub-gradient ascend [2], and step-adaptive sub-gradient ascent [7]. The number of iterations required by these algorithms is at least quadratically proportional to the inverse of the desired precision. In [8], an improved dual decomposition approach has been proposed for which the number of required iterations is proportional to the inverse of the desired precision. This means that the average number of iterations to achieve a precision of 1% is (a factor of) 100.

The optimality of the LDO for solving OSB has been shown in [2], [9], [10] basically by arguing that the duality gap is zero when the number of tones is large. Despite these results, finding a set of weight factors and Lagrange multipliers to solve (1) precisely is usually a tedious job and sometimes impossible in practice [8]. In fact, in [11] we showed that in some cases the LDO fails to find all points on the RR when there are no power constraints. This problem occurs when the RR is not strictly convex. In some extreme cases, the OSB solution provides significantly smaller bit-rates for some of the users than that can be obtained in theory or even by suboptimal techniques such as static spectrum management. To resolve this problem, iterative facet dividing (IFDA) algorithm has been proposed in [11]. It has been shown that the IFDA is capable of finding any point on the RR very closely. Moreover, the number of iterations is proportional to the logarithm of the inverse of the desired precision, meaning that to achieve a precision of 1% we only need (a factor of) $\log 100 \approx 4.6$ iterations.

978-1-61284-231-8/11/$26.00 ©2011 IEEE

The results in [11] are limited to the cases where there are no power constraints. Back to the general case, these results bring two key issues to attention. The first issue is whether there exist cases for which LDO is incapable of approaching points on the boundary of the RR. Stated in other words, are there any scenarios in which the RR is not strictly convex in the general case? The second issue is whether we can generalize IFDA to find Lagrange multipliers jointly with weight factors in order to take advantage of the robustness and speed of the technique. In this paper, we address these issues.

This paper is organized as follows. The multiuser DSL transmission system is described in Sec. II. In Sec. III, we extend the IFDA to the general case where there exists some power constraints. Simulation results are presented in Sec. IV, and finally the paper is concluded in Sec. V.

II. System Description

We assume N discrete multi-tone (DMT) DSL users with disjoint upstream (US) and downstream (DS) bands transmitting over K tones. In order to obtain K parallel MIMO channels, we assume that all users are DMT symbol synchronized at the receiver side. The bit-rate of user n per DMT symbol is

$$R_n = \sum_{k=1}^{K} b_k^{(n)} \tag{3}$$

where $b_k^{(n)}$ is the number of bits loaded to tone k for user n obtained by

$$b_k^{(n)} = \min\left\{b_{\max}, \left\lfloor \log_2\left(1 + \tfrac{1}{\Gamma}\mathrm{SNR}_k^{(n)}\right)\right\rfloor\right\}, \tag{4}$$

where $b_{\max}$ is the maximum number of bits that can be loaded to each tone, $\lfloor\cdot\rfloor$ denotes the floor function, Γ is the signal-to-noise ratio (SNR) gap, and $\mathrm{SNR}_k^{(n)}$ is the SNR at tone k of user n. The SNR is obtained by

$$\mathrm{SNR}_k^{(n)} = \frac{s_k^{(n)} g_k^{(n,n)}}{\sigma_k^{(n)} + \sum_{m=1;m\neq n}^{N} s_k^{(m)} g_k^{(n,m)}}, \tag{5}$$

where $s_k^{(n)}$ is the transmit PSD of user n at tone k, $\sigma_k^{(n)}$ is the PSD of the n-th receiver's noise at tone k, and $g_k^{(n,m)}$ is the channel's power gain from transmitter m to receiver n at tone k. In DSL systems, the transmit PSD of each user is bounded by a regulatory PSD mask, i.e., $s_k^{(n)} \leq s_{k,\mathrm{mask}}^{(n)}$. The aggregate transmit power of user n is $\Delta_{\mathrm{f}} P_n$ where Δ_{f} is the DMT tone spacing and

$$P_n \equiv \sum_{k=1}^{K} s_k^{(n)}. \tag{6}$$

The total transmit power that can be sent under the PSD mask is $\Delta_{\mathrm{f}} P_{\mathrm{mask}}^{(n)}$, where

$$P_{\mathrm{mask}}^{(n)} \equiv \sum_{k=1}^{K} s_{k,\mathrm{mask}}^{(n)}. \tag{7}$$

By substituting (3) and (6) into (2), the problem is decoupled into K parallel per-tone maximization problems, i.e.,

$$\underset{\{\boldsymbol{b}_k, \boldsymbol{s}_k\}}{\text{maximize}}\, \mathcal{L}_k; 1 \leq k \leq K, \tag{8}$$

where $\mathcal{L}_k \equiv \sum_{n=1}^{N} w_n b_k^{(n)} - \sum_{n=1}^{N} \lambda_n s_k^{(n)}$ is the per-tone Lagrangian on tone k. The per-tone problem is not convex in the general case and is solved by exhaustive search. However, nearly optimal algorithms exist for solving the per-tone problem with polynomial complexity in N. From (5), the element-wise minimum power vector $\boldsymbol{s}_k = \left(s_k^{(1)}, \ldots, s_k^{(N)}\right)^{\mathrm{T}}$ to load the bit-loading vector $\boldsymbol{b}_k = \left(b_k^{(1)}, \ldots, b_k^{(N)}\right)^{\mathrm{T}}$ on tone k is obtained by [1], [11]

$$\boldsymbol{s}_k = \left(\boldsymbol{D}_k - \Gamma \boldsymbol{B}_k \boldsymbol{C}_k\right)^{-1} \Gamma \boldsymbol{B}_k \boldsymbol{\sigma}_k, \tag{9}$$

where $\boldsymbol{D}_k = \mathrm{diag}\left\{g_k^{(1,1)}, \ldots, g_k^{(N,N)}\right\}$, $\boldsymbol{B}_k = \mathrm{diag}\left\{2^{\boldsymbol{b}_k}\right\} - \boldsymbol{I}_N$, $\boldsymbol{C}_k = \boldsymbol{G}_k - \boldsymbol{D}_k$ ($[\boldsymbol{G}_k]_{n,m} = g_k^{(n,m)}$), and $\boldsymbol{\sigma}_k = \left(\sigma_k^{(1)}, \ldots, \sigma_k^{(N)}\right)^{\mathrm{T}}$. The bit- and power-loading vector $\left(\boldsymbol{b}_k^{\mathrm{T}}, \boldsymbol{s}_k^{\mathrm{T}}\right)^{\mathrm{T}}$ is called achievable if $\boldsymbol{b}_k$ and $\boldsymbol{s}_k$ satisfy (9) and $0 \leq b_k^{(n)} \leq b_{\max}$ and $0 \leq s_k^{(n)} \leq s_{k,\mathrm{mask}}^{(n)}$ for $1 \leq n \leq N$.

III. Generalized IFDA

The IFDA as proposed in [11] is implemented using geometrical operations on the RR. To extend IFDA to the general case where the objective and constraint functions are defined on the bit rates as well as transmit powers, we consider the concept of joint rate and power region (JRPR). The JRPR on tone k, Φ_k, is the set of all achievable bit- and power-loading vectors $\left(\boldsymbol{b}_k^{\mathrm{T}}, \boldsymbol{s}_k^{\mathrm{T}}\right)^{\mathrm{T}}$. The JRPR over all tones is $\Phi = \Phi_1 \oplus \cdots \oplus \Phi_K$ where $\oplus$ denotes the Minkowski sum. The Minkowski sum of two sets A and B is defined by

$$A \oplus B \equiv \left\{\boldsymbol{a} + \boldsymbol{b} \middle| \forall \boldsymbol{a} \in A \wedge \boldsymbol{b} \in B\right\}. \tag{10}$$

If we look at R_n and P_n as independent variables and $\mathcal{L}$ as a constant, the Lagrangian $\mathcal{L} = \sum_{n=1}^{N} w_n R_n - \sum_{n=1}^{N} \lambda_n P_n$ can be interpreted as a hyperplane[1] $\mathcal{H}$ in the $2N$-D space with a normal vector $\boldsymbol{\nu} \equiv (w_1, w_2, \ldots, w_N, -\lambda_1, -\lambda_2, \ldots, -\lambda_N)^{\mathrm{T}}$. If we set $\mathcal{L} = \mathcal{L}_{\max}$ where $\mathcal{L}_{\max} \equiv \max_{\{\boldsymbol{b}_k, \boldsymbol{s}_k; \forall k\}} \sum_{n=1}^{N} w_n R_n - \sum_{n=1}^{N} \lambda_n P_n$, then $\mathcal{H}$ indicates a supporting hyperplane to Φ. Now consider the line $\ell : \left(t, R_{\min}^{(2)}, \ldots, R_{\min}^{(N)}, P_{\max}^{(1)}, \ldots, P_{\max}^{(N)}\right)$, where t is an independent variable. Assuming that the problem is feasible and all constraints in (1b) and (1c) are binding[2], the solution is located on ℓ. Moreover, when the boundary of the convex hull of Φ is in Φ, the solution is on the boundary of Φ. Therefore, the solution is located at the intersection of ℓ and the boundary of Φ and a supporting hyperplane to Φ

[1] We use a few N-D geometrical objects in this paper such as hyperplane, line, simplex, and facet. Please refer to [12] for their definitions.

[2] A constraint is called binding if it is met at equality in the solution, otherwise, it is unbinding or slack.

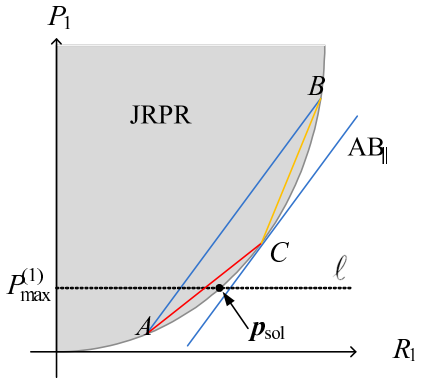

Fig. 1. An illustration of one iteration of the IFDA on the JRPR for a single-user scenario.

can be found at the solution. Figure 1 shows the JRPR (the shaded area) and ℓ (the dotted line) for a single user case[1]. In the following we use this figure to explain our algorithm.

Consider two points A and B on the boundary of the JRPR such that ℓ intersects $\overline{AB}$ (the line segment between A and B) at $\boldsymbol{p}_{\text{sol}}$. Note that $\boldsymbol{p}_{\text{sol}}$ is located on the boundary of Φ on the arc between the points A and B. Assume $\boldsymbol{\nu}_{\overline{AB}} = (\nu_1, \nu_2)^{\text{T}}$ is the normal vector to $\overline{AB}$ where $\nu_1 > 0$. Consider line $AB_{||}$, the supporting hyperplane to Φ parallel to $\overline{AB}$, and let $C = (R_C, P_C)^{\text{T}}$ denote the point at which $AB_{||}$ supports Φ. Since $AB_{||}$ is characterized by $\nu_1 R_1 + \nu_2 P_1 = \nu_1 R_C + \nu_2 P_C$, C can be reached by maximizing the Lagrangian $\mathcal{L} = w_1 R_1 - \lambda_1 P_1$ with $w_1 = \nu_1$ and $\lambda_1 = -\nu_2$. Now consider the line segments $\overline{AC}$ and $\overline{BC}$. Since ℓ intersects $\overline{AB}$, it intersects either $\overline{AC}$ or $\overline{BC}$. In Fig. 1, ℓ has intersected $\overline{AC}$. As it can be seen, the solution point $\boldsymbol{p}_{\text{sol}}$ is also between A and C on the boundary of Φ. As it can be realized, starting from $\overline{AB}$, we have found a shorter line segment $\overline{AC}$ intersected by ℓ with its ends located on the boundary of Φ. Therefore, by applying this procedure iteratively and assuming that the convex hull of JRPR is strictly convex, we would find two points in each iteration which eventually converge together at the solution.

The algorithm can be readily generalized to $N > 1$. For $N > 1$, the points A and B are replaced by $2N$ starting points $\boldsymbol{p}_1$ to $\boldsymbol{p}_{2N}$. The starting points indicate a facet F in $2N$ space which by assumption is intersected by ℓ[2]. To operate each iteration, the normal vector $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_{2N})^{\text{T}}$ (with $\nu_1 > 0$) to F is calculated and the Lagrangian is maximized by setting $w_1 = \nu_1$, …, $w_N = \nu_N$, $\lambda_1 = -\nu_{N+1}$, …, $\lambda_N = -\nu_{2N}$ to obtain a new point $\boldsymbol{q} = (R_1, \ldots, R_N, P_1, \ldots, P_N)^{\text{T}}$ on the boundary of Φ. Point $\boldsymbol{q}$ forms a facet with any $2N-1$ selection of points from $\boldsymbol{p}_1$ to $\boldsymbol{p}_{2N}$. Let F_n denote the facet indicated by the points $\boldsymbol{q}, \boldsymbol{p}_1, \ldots, \boldsymbol{p}_{n-1}, \boldsymbol{p}_{n+1}, \ldots, \boldsymbol{p}_{2N}$. Since ℓ intersects F, it intersects one of the facets F_1 to F_{2N} as well. Let F_m denote the intersected facet. Then the vertices of F_m, i. e., $\boldsymbol{q}, \boldsymbol{p}_1, \ldots, \boldsymbol{p}_{m-1}, \boldsymbol{p}_{m+1}, \ldots, \boldsymbol{p}_{2N}$, are the starting points for the next iteration of the algorithm.

[1]The JRPR cannot be illustrated for $N > 1$, as it has at least 4 dimensions in that case.

[2]We will propose an algorithm for finding the starting points in Sec. III-B.

The algorithm is terminated when the required precision is achieved or when the last calculated point is located on the last facet. The second case could only happen when the JRPR is not strictly convex. In this case, traditional LDO based algorithms would fail to find the solution. However, in Sec. III-C, we propose a mixing algorithm which finds the solution using the bit- and power-loadings associated with the vertices of the last facet.

A. Unbinding Power Constraints and Negative Lagrange Multipliers

Usually, the weight factors and Lagrange multipliers associated with inequality constraints are considered to be non-negative. The IFDA sets the weight factors and Lagrange multipliers according to the elements of the normal vector to a facet which may not necessarily satisfy this condition. More explicitly, as $\nu_1 = w_1$ is enforced to be positive, $\nu_2 = w_2$ to $\nu_N + w_N$ should be non-negative, and $\nu_{N+1} = -\lambda_1$ to $\nu_{2N} = -\lambda_N$ should be non-positive for all iterations. Our simulation results show that sometimes one (or more) element(s) in ν_{N+1} to ν_{2N} take positive values leading to one (or more) negative Lagrange multiplier(s). This happens when a power constraint in (1c) is not binding (or the optimal point is close to a region where a power constraint is not binding). To resolve this issue, λ_n are let to take negative values by modifying the Lagrangian $\mathcal{L}$ and per-tone Lagrangians $\mathcal{L}_k$ as follows:

$$\tilde{\mathcal{L}} \equiv \sum_{n=1}^{N} w_n R_n - \sum_{n=1}^{N} \lambda_n \tilde{P}_n, \tag{11}$$

$$\tilde{\mathcal{L}}_k \equiv \sum_{n=1}^{N} w_n b_k^{(n)} - \sum_{n=1}^{N} \lambda_n \tilde{s}_k^{(n)}, \tag{12}$$

where $\tilde{P}_n \equiv \begin{cases} P_n; & \lambda_n \geq 0 \\ P_{\max}^{(n)}; & \lambda_n < 0 \end{cases}$ and $\tilde{s}_n \equiv \begin{cases} s_n; & \lambda_n \geq 0 \\ s_{\max}^{(n)}; & \lambda_n < 0 \end{cases}$. When λ_n is positive, $\tilde{\mathcal{L}}$ and $\tilde{\mathcal{L}}_k$ reduce to $\mathcal{L}$ and $\mathcal{L}_k$. However, when λ_n is negative, the power of user n is virtually assumed to be equal to the maximum value in the range[3]. However, the actual power sent by user n is independently set during the exhaustive search for finding the optimal point and can be smaller. This avoids bit rate loss for other users by experiencing less crosstalk induced by $s_k^{(n)}$. Note that the point $\boldsymbol{q}$ calculated in each iteration of the IFDA is redefined as well by $\boldsymbol{q} = \left(R_1, \ldots, R_N, \tilde{P}_1, \ldots, \tilde{P}_N\right)^{\text{T}}$.

From the geometrical point of view this modification is equal to adding all $2^N - 1$ projections of Φ on the N hyperplanes $P_n = P_{\text{mask}}^{(n)}$ and their intersections to Φ. By this modification ℓ intersects Φ even if the power constraints are not binding.

[3]In the general case, we might not have a PSD mask limitation. However, due to practical limitations, the transmit power on each tone is limited to a maximum value which can be used instead of $s_{k,\text{mask}}^{(n)}$ in (12).

B. The Expanding Algorithm for Finding the Starting Points

The generalized IFDA discussed above requires $2N$ starting points $\boldsymbol{p}_1$ to $\boldsymbol{p}_{2N}$ at the start-up. Here we propose an algorithm which finds the starting points for the algorithm.

If the problem is feasible then the point $\boldsymbol{p}_\circ \equiv \left(0, R_{\min}^{(2)}, \ldots, R_{\min}^{(N)}, P_{\max}^{(1)}, \ldots, P_{\max}^{(N)}\right)^{\mathrm{T}} \in \ell$ is located on the boundary of the convex hull of the JRPR. The algorithm works by making a $2N$-D simplex S_{expan} which is iteratively expanded toward $\boldsymbol{p}_\circ$. Since ℓ passes through $\boldsymbol{p}_\circ$, it is guaranteed that ℓ intersects S_{expan} after a few iterations. The algorithm, called the expanding algorithm, is explained in more details in the following:

1) First $2N$ distinct points $\boldsymbol{r}_1$ to $\boldsymbol{r}_{2N}$ are calculated by solving the problem at $2N$ different arbitrarily (e.g. randomly) selected vectors of $(w_1, \ldots, w_N, \lambda_1, \ldots, \lambda_N)$. Let $F_\circ$ denote the facet indicated by $\boldsymbol{r}_1$ to $\boldsymbol{r}_{2N}$.
2) If ℓ intersects $F_\circ$, then $\boldsymbol{r}_1$ to $\boldsymbol{r}_{2N}$ can be used as the starting points. Otherwise, we go to Step 3.
3) A new point is calculated by solving the problem at $(w_1, \ldots, w_N, \lambda_1, \ldots, \lambda_N)^{\mathrm{T}} = \boldsymbol{\nu}$, where $\boldsymbol{\nu}$ is a normal vector to $F_\circ$. The sign of $\boldsymbol{\nu}$ should be selected such that its direction is towards $\boldsymbol{p}_\circ$. Let S_{expan} denote the simplex identified by the vertices $\boldsymbol{p}_\circ$ and $\boldsymbol{r}_1$ to $\boldsymbol{r}_{2N}$.
4) If ℓ intersects S_{expan} then it intersects two facets of S_{expan}. The vertices of either of these facets can be used as the starting points $\boldsymbol{p}_1$ to $\boldsymbol{p}_{2N}$, however, for faster convergence, the facet intersected by ℓ at a greater value of $t(= R_1)$ is preferable.
5) If ℓ does not intersect S_{expan}, then we can find at least one facet of S_{expan}, namely F_1, for which $\boldsymbol{p}_\circ$ is located on one side of it and the remaining vertices of S_{expan} are located on the other side of it. Now we replace $\boldsymbol{r}_1$ to $\boldsymbol{r}_{2N}$ by the vertices of F_1 and $F_\circ$ by F_1 and return to Step 3.

C. The Mixing Algorithm

When the solution is located on a nonstrictly-convex region of the JRPR boundary, it is usually not achievable by a particular set of weight factors and Lagrange multipliers. Fortunately, the generalized IFDA provides us with $2N$ points surrounding the solution. By using the per-tone bit- and power-loading vectors associated with these points in a mixed fashion, we can reach the solution very closely when the number of tones is large.

Assume the final points $\boldsymbol{p}_1$ to $\boldsymbol{p}_{2N}$ indicate facet F_{final} on the boundary of the JRPR and let $\boldsymbol{b}_{k,m} = (b_{k,m}, \ldots, b_{k,m})^{\mathrm{T}}$ and $\boldsymbol{s}_{k,m} = (s_{k,m}, \ldots, s_{k,m})^{\mathrm{T}}$ denote the bit- and power-loading vectors associated with $\boldsymbol{p}_m$ on tone k. Let $\lambda_{n,m}$ denote the n-th Lagrange multiplier associated with the m-th point. We define $\tilde{\boldsymbol{s}}_{k,m} = \left(\tilde{s}_{k,m}^{(1)}, \ldots, \tilde{s}_{k,m}^{(N)}\right)^{\mathrm{T}}$ where $\tilde{s}_{k,m}^{(n)} = \begin{cases} s_{k,m}^{(n)}, & \lambda_{n,m} \geq 0 \\ s_{k,\text{mask}}^{(n)}, & \lambda_{n,m} < 0 \end{cases}$. Note that $\boldsymbol{p}_m = \left(\sum_{k=1}^{K} [\boldsymbol{b}_{k,m}]_1, \ldots, \sum_{k=1}^{K} [\boldsymbol{b}_{k,m}]_N, \sum_{k=1}^{K} [\tilde{\boldsymbol{s}}_{k,m}]_1, \ldots, \sum_{k=1}^{K} [\tilde{\boldsymbol{s}}_{k,m}]_N\right)^{\mathrm{T}}$, where $[\boldsymbol{x}]_n$ denotes the n-th element of vector $\boldsymbol{x}$. Let $\boldsymbol{p}_{\text{sol}} = \left(R_{\min}^{(1)}, \ldots, R_{\min}^{(N)}, P_{\max}^{(1)}, \ldots, P_{\max}^{(N)}\right)^{\mathrm{T}}$ denote the intersection point of ℓ and F_{final}. Let $\mu_k \in [1 : 2N]$ indicate the point whose bit- and power-loadings will be used on tone k. The sequence μ_1 to μ_K indicate a point on F_{final} calculated by $\boldsymbol{p}_{\text{mix}} = \left(\sum_{k=1}^{K} [\boldsymbol{b}_{k,\mu_k}], \ \ldots, \sum_{k=1}^{K} [\boldsymbol{b}_{k,\mu_k}]_N, \sum_{k=1}^{K} [\tilde{\boldsymbol{s}}_{k,\mu_k}]_1, \ \ldots, \sum_{k=1}^{K} [\tilde{\boldsymbol{s}}_{k,\mu_k}]_N\right)^{\mathrm{T}}$. The goal is to optimize μ_1 to μ_K such that $\boldsymbol{p}_{\text{mix}}$ is located close to $\boldsymbol{p}_{\text{sol}}$ within the required precision. Algorithm 1 can be used for this purpose.

Algorithm 1: The Mixing Algorithm

```
/* STEP 1                                                    */
Set p_mix ← p_1 and μ_k ← 1 for 1 ≤ k ≤ K;
/* STEP 2                                                    */
repeat
    for k = 1 ... K do
        p_mix ← p_mix − (b^T_{k,μ_k}, s̃^T_{k,μ_k})^T;
        μ_k ← index_m min dist { p_sol, p_mix + (b^T_{k,m}, s̃^T_{k,m})^T };
        p_mix ← p_mix + (b^T_{k,μ_k}, s̃^T_{k,μ_k})^T;
until no improvements can be made;
/* STEP 3                                                    */
repeat
    Pick two random tone indices k_1, k_2; k_1 ≠ k_2;
    p_mix ← p_mix
        − (b^T_{k_1,μ_{k_1}}, s̃^T_{k_1,μ_{k_1}})^T − (b^T_{k_2,μ_{k_2}}, s̃^T_{k_2,μ_{k_2}})^T;
    (μ_{k_1}, μ_{k_2}) ← index_{(m_1,m_2)} min dist { p_sol, p_mix
        + (b^T_{k_1,m_1}, s̃^T_{k_1,m_1})^T + (b^T_{k_2,m_2}, s̃^T_{k_2,m_2})^T };
    p_mix ← p_mix
        + (b^T_{k_1,μ_{k_1}}, s̃^T_{k_1,μ_{k_1}})^T + (b^T_{k_2,μ_{k_2}}, s̃^T_{k_2,μ_{k_2}})^T.
until the desired precision achieved or the maximum
number of iterations is reached;
```

The algorithm works as follows: In the first step we assume that $\boldsymbol{p}_{\text{mix}}$ is equal to $\boldsymbol{p}_1$ and we set all indices μ_k equal to 1. In the second step, we repeatedly test all tones k and find the point μ_k which minimizes the normalized Euclidean distance of $\boldsymbol{p}_{\text{mix}}$ and $\boldsymbol{p}_{\text{sol}}$ if its associated bit- and power-loadings is selected on tone k. The normalized Euclidean distance of $\boldsymbol{p}_{\text{mix}}$ and $\boldsymbol{p}_{\text{sol}}$ is calculated by normalizing the first N dimensions by $R_{\min}^{(1)}$ to $R_{\min}^{(N)}$ and the second N dimensions by $P_{\max}^{(1)}$ to $P_{\max}^{(N)}$. Finally, in the third step, we pick two random tones k_1 and k_2 and find the points μ_{k_1} and μ_{k_2} minimizing the distance of $\boldsymbol{p}_{\text{mix}}$ to $\boldsymbol{p}_{\text{sol}}$ if their bit- and power- loadings are selected on k_1 and k_2, respectively. This procedure is repeated until we reach the desired precision or the maximum number of iterations.

TABLE I
SIMULATION PARAMETERS

PARAMETER	VALUE
Crosstalk model	ANSI standard 1% worst-case model [13] without considering FSAN power sum rule
Bandplan and PSD mask	VDSL 997 (DS: .138-3, 5.1-7.05 MHz, US: 3-5.1, 7.05-12 MHz) [14] VDSL2E17 B7-9 (DS: .138-3, 5.1-7.05, 12-14 MHz, US: 3-5.1, 7.05-12, 14-17.67 MHz) [15]
Cable type	26 AWG [13]
Noise	White noise, -140 dBm/Hz
Tone spacing, Δ_{f}	4.3125 kHz
Symbol rate, f_{s}	4 kHz
$b_{\max}$	15
$b_{\min}$	2
SNR gap, Γ	12.0 dB
Per tone PSD granularity	3 dB

IV. SIMULATION RESULTS

For the first simulation, we consider a two-user US VDSL scenario in which two equal length loops are transmitting over a 250 m 26-AWG cable. We set $P_{\max}^{(n)} = .2 \times P_{\mathrm{mask}}^{(n)}$ = 1.2058 mW and $R_{\min}^{(2)}$ = 52 Mbps. Other simulation parameters are listed in Table I.

The expanding algorithm converges after two iterations, and then the generalized IFDA reaches the optimal point in 32 iterations with precision 10^{-16}. The vertices of the final facet are

$\boldsymbol{p}_1 = (34.91\mathrm{Mbps}, 69.01\mathrm{Mbps},\ .79\mathrm{mW}, 1.61\mathrm{mW})^{\mathrm{T}}$,
$\boldsymbol{p}_2 = (81.22\mathrm{Mbps}, 22.71\mathrm{Mbps}, 2.27\mathrm{mW},\ .14\mathrm{mW})^{\mathrm{T}}$,
$\boldsymbol{p}_3 = (22.71\mathrm{Mbps}, 81.22\mathrm{Mbps},\ .14\mathrm{mW}, 2.27\mathrm{mW})^{\mathrm{T}}$, and
$\boldsymbol{p}_4 = (58.44\mathrm{Mbps}, 45.49\mathrm{Mbps}, 1.20\mathrm{mW}, 1.21\mathrm{mW})^{\mathrm{T}}$.

The corresponding vectors of weight factors and Lagrange multipliers ($\boldsymbol{\omega}_m \equiv (w_{1,m}, \ldots, w_{N,m}, \lambda_{1,m}, \ldots, \lambda_{N,m})^{\mathrm{T}}$) are

$\boldsymbol{\omega}_1 = (.7058469, .7058471, .0421898, .0421901)^{\mathrm{T}}$,
$\boldsymbol{\omega}_2 = (.7058475, .7058475, .0421811, .0421811)^{\mathrm{T}}$,
$\boldsymbol{\omega}_3 = (.7058474, .7058474, .0421826, .0421826)^{\mathrm{T}}$, and
$\boldsymbol{\omega}_4 = (.7058463, .7058461, .0422038, .0422034)^{\mathrm{T}}$.

As it can be seen, the obtained points are located relatively far away from each other on the boundary of the JRPR. However, the vectors of weight factors and Lagrange multipliers are very close to each other. This clearly shows that the boundary of the JRPR is not strictly convex and is flat at least to a precision of 10^{-4}. As a result, a very small change in the weight factors or Lagrange multipliers causes a large change in the obtained solution which explains why traditional algorithms for tuning the weight factors and Lagrange multipliers fail to find the solution. Line ℓ intersects the facet indicated by $\boldsymbol{p}_1$ to $\boldsymbol{p}_4$ at $\boldsymbol{p}_{\mathrm{sol}} = (51.9272\mathrm{Mbps}, 52.0000\mathrm{Mbps}, 1.2058\mathrm{mW},\ 1.2058\mathrm{mW})^{\mathrm{T}}$. By mixing the bit- and power-loadings of $\boldsymbol{p}_1$ to $\boldsymbol{p}_4$ using the proposed mixing algorithm, we obtain $\boldsymbol{p}_{\mathrm{mix}} = (51.9302\mathrm{Mbps}, 51.9981\mathrm{Mbps}, 1.2060\mathrm{mW}, 1.2060\mathrm{mW})^{\mathrm{T}}$. As it can be seen, $\boldsymbol{p}_{\mathrm{mix}}$ is considerably close to $\boldsymbol{p}_{\mathrm{sol}}$.

Figure 2 compares the convergence properties of the proposed algorithm with the step-adaptive subgradient method proposed in [7]. Four curves have been plotted in this figure. The solid curve shows the (normalized Euclidean) distance of the resulting point to the target point for the subgradient algorithm vs. the number of iterations. The original step-adaptive subgradient algorithm described in [7] works by finding the Lagrange multipliers after fixing the weight factors as the RR is unknown before solving the problem. This means that the weight factors have to be optimized separately leading to considerably higher number of iterations. Here, we know the location of the solution on the RR, and we have used it to find the weight factors and the Lagrange multipliers simultaneously. The other three curves in Fig. 2 show the results for the generalized IFDA. The dotted curve shows the distance of the last obtained point in each iteration with the solution. The green curve shows the distance of the last obtained point with the corresponding facet. Finally, the dash-dotted curve shows the distance of the solution with the intersection point of ℓ and the last facet.

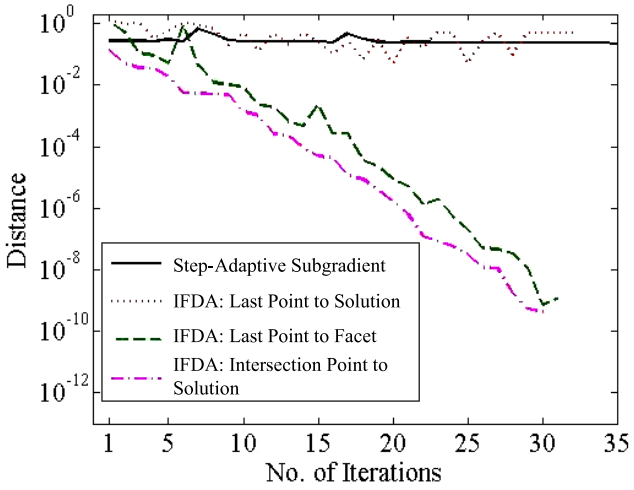

Fig. 2. Convergence of the proposed algorithm compared to the step-adaptive subgradient algorithm for a two-user US VDSL scenario.

The subgradient algorithm converges in about 80 iterations. The minimum achieved distance to the solution for the algorithm is .2185. The distance of the points obtained by the generalized IFDA to ℓ is on the same order of magnitude. However, as it can be seen, the distance of the intersection point to the solution decreases with an almost constant slope in the logarithmic scale meaning that the required number of iterations is logarithmic in the inverse of the desired precision[1].

These observations also imply that the solution is located on a facet on the boundary of the JRPR. As a result, the solution is not achievable by tuning the weight factors and Lagrange multipliers. That is because only one of the points located on a facet of the JRPR can be reached by setting the weight factors and Lagrange multipliers according to the normal vector of

[1]We have analytically proved that the required number of iterations for the algorithm is logarithmic in the inverse of the desired precision which is not included here due to lack of space.

that facet [11]. On the other hand, the generalized IFDA is capable of reaching the point as it mixes the bit-loadings and politeness values for $2N$ points located around the target point on the final facet.

As for the second simulation, we consider a case in which the power constraints are not binding for a user. This scenario is a near-far two-user US VDSL2 scenario with one user located at 100 m and the other user located at 500 m from the central office. For this scenario, we set $P_{\max}^{(n)} = .2 \times P_{\text{mask}}^{(n)}$ and $R_{\min}^{(2)} = 49$ Mbps, where $P_{\text{mask}}^{(n)} = 35.90$ mW. The desired precision is 1%. By executing the algorithm, the vertices of the final facet are obtained as

$\boldsymbol{p}_1 = (94.954\text{Mbps}, 53.388\text{Mbps}, \ .36\text{mW}, 13.28\text{mW})^{\text{T}}$,

$\boldsymbol{p}_2 = (102.490\text{Mbps}, 46.074\text{Mbps}, \ .36\text{mW}, 4.31\text{mW})^{\text{T}}$,

$\boldsymbol{p}_3 = (102.490\text{Mbps}, 48.268\text{Mbps}, \ .36\text{mW}, 8.26\text{mW})^{\text{T}}$, and

$\boldsymbol{p}_4 = (102.490\text{Mbps}, 46.806\text{Mbps}, \underline{35.90\text{mW}}, 4.67\text{mW})^{\text{T}}$.

The corresponding vectors of weight factors and Lagrange multipliers are

$\boldsymbol{\omega}_1 = (.703561, .709315, .00241426, .0432261)^{\text{T}}$,

$\boldsymbol{\omega}_2 = (.636459, .679566, .0856679, .354642)^{\text{T}}$,

$\boldsymbol{\omega}_3 = (.735414, .663412, .00127636, .138016)^{\text{T}}$, and

$\boldsymbol{\omega}_4 = (.635729, .658192, \underline{-.261314}, .307158)^{\text{T}}$.

As it can be seen, the first Lagrange multiplier for the fourth point (i.e., the third element of $\boldsymbol{\omega}_4$) is negative, which has resulted in a full virtual transmit power for the first user (the third element of $\boldsymbol{p}_4$). Using the mixing algorithm, we obtain

$\boldsymbol{p}_{\text{mix}} = (101.0\text{Mbps}, 49.0\text{Mbps}, \ .36\text{mW}, 7.18\text{mW})^{\text{T}}$,

which indicates that $R_2 = R_{\min}^{(2)}$, $P_1 < P_{\max}^{(1)}$, and $P_2 = P_{\max}^{(2)}$. As it can be seen, the algorithm has successfully identified the unbinding constraint (the power constraint for the first user) and the binding constraints (the bit rate and power constraints for the second user) and is capable of solving the problem for the cases in which some constraints are not binding without any prior knowledge about the binding and unbinding constraints.

V. Conclusion

Dual optimization techniques are optimal for spectrum balancing of DSL, merely when the JRPR is strictly convex. When the JRPR is not strictly convex, we may not be able to find the desired solution by tuning the weight factors and Lagrange multipliers. We proposed a new algorithm which finds the optimal solution by mixing the solution of the dual problem for $2N$ points surrounding the target point. The algorithm consists of three parts: the expanding algorithm which finds the starting points, the generalized IFDA, and the mixing algorithm. The algorithm is capable of finding the optimal solution without any prior knowledge about binding and unbinding power constraints. Our simulation results show that the number of iterations required for the algorithm to converge is proportional to the logarithm of the inverse of the desired precision.

References

[1] R. Cendrillon, W. Yu, M. Moonen, J. Verlinden, and T. Bostoen, "Optimal multiuser spectrum balancing for digital subscriber lines," *IEEE Trans. Commun.*, vol. 54, no. 5, pp. 922 – 933, May 2006.

[2] W. Yu and R. Lui, "Dual methods for nonconvex spectrum optimization of multicarrier systems," *IEEE Trans. Commun.*, vol. 54, no. 7, pp. 1310 –1322, Jul. 2006.

[3] M. Mohseni, R. Zhang, and J. Cioffi, "Optimized transmission for fading multiple-access and broadcast channels with multiple antennas," *IEEE J. Select. Areas Commun.*, vol. 24, no. 8, pp. 1627 –1639, Aug. 2006.

[4] A. Marques, X. Wang, and G. Giannakis, "Optimal stochastic dual resource allocation for cognitive radios based on quantized CSI," in *IEEE Int'l. Conf. on Acoustics, Speech, & Signal Processing, ICASSP'08*, Las Vegas, NA, Mar. 2008, pp. 2801 –2804.

[5] L. Xiao, M. Johansson, and S. Boyd, "Simultaneous routing and resource allocation via dual decomposition," *IEEE Trans. Commun.*, vol. 52, no. 7, pp. 1136 – 1144, Jul. 2004.

[6] W. Yu, "Sum-capacity computation for the gaussian vector broadcast channel via dual decomposition," *IEEE Trans. Inform. Theory*, vol. 52, no. 2, pp. 754 –759, Feb. 2006.

[7] P. Tsiaflakis, J. Vangorp, M. Moonen, and J. Verlinden, "A low complexity optimal spectrum balancing algorithm for digital subscriber lines," *Signal Process*, vol. 87, no. 7, pp. 1735–1753, Jul. 2007.

[8] P. Tsiaflakis, I. Necoara, J. Suykens, and M. Moonen, "Improved dual decomposition based optimization for DSL dynamic spectrum management," *IEEE Trans. Signal Processing*, vol. 58, no. 4, pp. 2230–2245, Apr. 2010.

[9] Z.-Q. Luo and S. Zhang, "Dynamic spectrum management: complexity and duality," *IEEE J. Select. Topics Signal Processing*, vol. 2, no. 1, pp. 57–72, Feb. 2008.

[10] ——, "Duality gap estimation and polynomial time approximation for optimal spectrum management," *IEEE Trans. Signal Processing*, vol. 57, no. 7, pp. 2675–2689, Jul. 2009.

[11] A. R. Forouzan, "Optimal spectrum management of DSL with non-strictly convex rate region," *IEEE Trans. Signal Processing*, vol. 57, no. 7, pp. 2558 –2568, Jul. 2009.

[12] E. W. Weisstein, "Mathword," A Wolfram web resource. [Online]. Available: http://mathworld.wolfram.com/

[13] "Spectrum management for loop transmission systems," ANSI Standard T1.417-2003, Feb. 2003.

[14] ETSI, "Transmission and Multiplexing (TM); Access transmission systems on metallic access cables; Very High Speed Digital Subscriber Line (VDSL); Part I: Functional Requirements," ETSI Std. TS 101 270-1, Rev. V.1.3.1, 2003.

[15] ITU-T G.993.2; Amendment 1, "Very high speed digital subscriber line transceivers 2 (VDSL2)," Geneva, Switzerland, Apr. 2007.