Enhancement of the Iterative Spectrum Balancing Algorithm for Power Allocation in DSL Systems

Ali Kalakech, Jérôme Louveaux, Luc Vandendorpe
Communications and Remote Sensing Laboratory, Ecole polytechnique de Louvain
place du Levant 2, B-1348 Louvain-la-Neuve, Belgium
Email: {ali.kalakech, jerome.louveaux, luc.vandendorpe}@uclouvain.be

Abstract—The optimization of power allocation in DSL systems is a well studied non-convex problem. In recent years many algorithms have been proposed to solve this problem, but due to the non-convexity, researchers had to rely on heuristic, or exhaustive search algorithms. Iterative spectrum balancing (ISB) is one of these algorithms, that relies on iterative exhaustive search over individual power to achieve the optimization. In this paper we propose to take advantage of an approximate gradient model to reduce the complexity of ISB. Furthermore techniques such as the additional starting point (ASP) and successive optimization are used to bring the result closer to the global optimum.

I. INTRODUCTION

One of the major limitations in DSL systems is crosstalk. Crosstalk is the electromagnetic coupling between twisted pairs that creates interference between the different lines in DSL systems. In VDSL systems it is the dominating noise factor. A user transmitting with an excessive power in a given binder can deteriorate all the lines of that binder.

Operators and standardization bodies imposed limitations on the total power and on the PSD (power spectral density) mask of each system. PSD masks help to limit the level of crosstalk. However these PSD masks are based on the worst case scenario. Hence power allocations following these masks are far from optimal in practical situations.

To improve the power allocation in DSL systems, dynamic spectrum management (DSM) was introduced [1]. DSM tries to optimize users’ bit rates by adapting their power and spectral shapes to the channel loss and the ambient crosstalk across frequencies. DSM algorithms can be classified into two categories: distributed algorithms and centralized algorithms.

One of the first distributed algorithms is Iterative WaterFilling (IWF) [2]. In this algorithm each modem tries to optimize its own bit rate with respect to the ambient noise, including the current crosstalk. The optimization is done using the classical water filling algorithm. This procedure is repeated iteratively on all modems until a constant level of power and crosstalk is reached for all users.

Autonomous Spectrum Balancing (ASB) [3] is another distributed algorithm where each modem tries to optimize the total sum of its own bit rate and the bit rate of a virtual line. The virtual line is supposed to represent a typical weak line, thus limiting the effect of each modem on its neighbors.

Centralized algorithms propose the establishment of a system management center (SMC). In the SMC the total knowledge of the channel gains (both crosstalk channels and direct ones) is supposed to be given which allows the optimization of the entire system. Due to interference, the optimization of the DSL system total capacity is an NP hard non convex problem.

OSB (optimal spectrum balancing) [4] was proposed as the optimal centralized algorithm. OSB makes use of the Lagrange function to include the power constraints in the objective function. This enables the optimization to be carried out per tone. To find the optimal solution at each tone an exhaustive search over all the different possible power allocations is implemented.

In [5] the use of a Lagrangian to relax the power constraints is justified and the Iterative Spectrum Balancing (ISB) is introduced. The ISB algorithm ([5], [6]) replaces the exhaustive search over all the possible power allocations by a simple exhaustive line search algorithm over individual users’ power. This procedure reduces the complexity of OSB. The SCALE algorithm [7] and others [8] resort to a relaxation of the objective function which allows the use of convex optimization.

This paper is based on the iterative per-user optimization used in ISB. The per-user optimization step is here simplified by finding the roots of an approximate gradient model instead of the exhaustive line search. The ISB algorithm is only guaranteed to reach a local optimum. In order to increase the performance we propose to use the additional starting point technique. In combination with the above simplification, the complexity is low and the performance obtained is better.

The use of different initial points to perform the optimization might however result in a perturbed and non continuous power allocation over the tones. We propose to combine successive optimization and ASP, which provides a smoothening effect and improves the optimization further.

In section II we formulate the optimization problem and give a review of the ISB algorithm. In section III we show that the gradient over individual users can be approximated by a hyperbolic model. In section IV we present the combination of successive optimization and ASP. Section V reports the numerical results. Finally we conclude in section VI.
II. System Model, OSB and ISB Review

In this section we define the model used for the DSL system. From this model the optimization problem is formulated and a review of the solutions proposed in the literature (OSB and ISB algorithms) is given.

A. System Model

We consider DSL systems using DMT (discrete multiple tone). Assuming a proper use of the cyclic prefix technique, the channel may be decomposed in $N$ parallel subchannels. We denote by $N$ the total number of tones/subchannels and by $K$, the total number of users. For each particular tone the system can be viewed as an interference channel. Viewing the interference from other users due to crosstalk as additive noise the capacity of user $i$ in sub-channel $n$ is given by Shannon formula:

$$R_i = B \sum_{n=1}^{N} \log_2(1 + \frac{1}{\Gamma} SNR_i(n)),$$

where $\Gamma$ is the SNR gap, $SNR_i(n)$ is given by

$$SNR_i(n) = \frac{|H_{ii}(n)|^2 P_i(n)}{\sigma^2_i(n) + \sum_{i \neq j} |H_{ij}(n)|^2 P_i(n)}.$$

The $H_{ii}(n)$ are the direct channel gains of user $i$ at tone $n$. $H_{ij}(n)$ is the crosstalk gain from line $l$ to line $i$ at tone $n$, and $P_i(n)$ is the power transmitted by line $i$. The background noise variance of user $i$ at tone $n$ is denoted by $\sigma_i^2(n)$. It is assumed that a PSD mask and a total power limitation are imposed to each user, so the problem may be stated as:

$$\text{max}_P \sum_{i=1}^{K} \omega_i R_i$$

Subject to

$$\sum_{n=1}^{N} P_i(n) \leq P_t$$

$$P_i(n) \in [0, P_{\text{max}}(n)]$$

where $P$ is a $N \times K$ matrix where each row correspond to the the power allocation at a tone $n$; $P(n) = [P_1(n), P_2(n) ... P_K(n)]$. The weighting coefficients $\omega_i$ are fixed parameters in the problem and supposed to be defined by external considerations on user’s priorities.

B. Review of OSB and ISB

Due to the presence of crosstalk, the objective function in (3) can be seen as a difference of two log functions which yields an NP hard non convex problem. Paper [4] tries to solve this problem globally by introducing the dual function to relax the power constraints. Paper [5] shows that problem (3) satisfies the time sharing condition. The authors also illustrate that problem (3) has zero duality gap and thus justify the use of the duality to solve it. The Lagrange function of problem (3) is given by:

$$g(\lambda, P) = \sum_{i}^{K} \omega_i R_i - \lambda_i \sum_{n=1}^{N} P_i(n) - P_t$$

where $\lambda = [\lambda_1, ..., \lambda_i, ..., \lambda_K]$ and each $\lambda_i$ represents a Lagrangian multiplier. For fixed $\lambda$, the function $f$ is defined as:

$$f(\lambda) = \max_P g(\lambda, P).$$

The dual problem becomes:

$$\min_{\lambda} f(\lambda)$$

Subject to $\lambda_i \geq 0.$

To solve problem (6) both OSB and ISB propose a double loop iterative procedure. An outer loop searches for the appropriate $\lambda$ that minimizes $f(\lambda)$ to meet the power constraints. And for each set of fixed $\lambda_i$, an inner loop maximizes $g(\lambda, P)$ with respect to $P$. A simple gradient algorithm is sufficient for finding $\lambda$ in the outer loop, but this is not the case with the inner loop optimization which is still a non-convex problem. Examining $g(\lambda, P)$ shows that for fixed $\lambda$ there is no coupling between the tones $n$ as $\sum_{i}^{K} \lambda_i P_i$ becomes a constant which no longer affects the optimization. Hence the optimizations can be carried out per tone. This makes the complexity linear in function of $N$. For each tone, the non-convexity property holds. So, to solve the tone wise optimization, OSB has been proposed where an exhaustive search over all the possible power allocations in a tone $n$ is implemented. This renders the complexity exponential with $K$.

To reduce the complexity, ISB has been put forward. An exhaustive “line search” is performed over the power of individual users instead of a total exhaustive search. For each tone $n$, the power of $(K-1)$ users is fixed and an exhaustive search is performed over the power $P_k(n)$ of the remaining user to maximize $g(\lambda, P)$. This procedure is repeated iteratively over all users till a constant power allocation is reached. In the next section we propose a simple algorithm based on the gradient to replace the exhaustive line search proposed in ISB. This reduces the ISB complexity further.

III. Complexity Reduction

In this section we describe an algorithm that maximizes the objective function $g(\lambda, P)$ at tone $n$ with respect to $P_k(n)$. The power allocation at the other $K-1$ users is considered constant. This algorithm tries to find directly the optimum by searching the critical points at which the gradient equal to zero. This algorithm replaces the exhaustive line search proposed in ISB for the inner loop. It reduces the complexity considerably.

A. Partial Derivative Approximation

A simple way to perform the optimization of $g(\lambda, P)$ at tone $n$ with respect to $P_k(n)$ is to find all the roots of the derivatives.

$$\frac{\partial g(\lambda, P)}{\partial P_k(n)} = 0.$$
The partial derivative of $g(\lambda, \mathbf{P})$ with respect to $P_k(n)$ is given by
\[
\frac{\partial g(\lambda, \mathbf{P})}{\partial P_k(n)} = \frac{\omega_k}{P_k(n) + C_k} + \sum_{i \neq k} \frac{\omega_i}{P_k(n) + C_i},
\]
where $C_k, C_1, C_2$ are defined as follows:
\[
C_k = \frac{\sum_{i \neq k} |H_{ki}(n)|^2 P_i(n) + \sigma^2}{|H_{kk}(n)|^2},
\]
\[
C_1 = \frac{\sum_{i \neq k} |H_{ki}(n)|^2 P_i(n) + \sigma^2}{|H_{kk}(n)|^2},
\]
\[
C_2 = \frac{\sum_{i \neq k, i} |H_{ki}(n)|^2 P_i(n) + \sigma^2}{|H_{kk}(n)|^2}.
\]

We notice that $\frac{\partial g(\lambda, \mathbf{P})}{\partial P_k(n)}$ is the sum of different hyperbolas of the form $y = \frac{a}{P_k(n) + c}$, where $-C_k, -C_1, -C_2$ represent the different asymptotes of these hyperbolas. These asymptotes are all negative. Since the direct channel gain $|H_{kk}(n)|^2$ is much bigger than the crosstalk channel gains $|H_{ik}(n)|^2$ we can conclude that $C_k$ is much smaller than the other asymptotes. So for $0 < P_k(n) < P_{\text{max}}(n)$ and when $P_{\text{max}}$ is low, the partial derivative $\frac{\partial g(\lambda, \mathbf{P})}{\partial P_k(n)}$ is mainly influenced by the hyperbola corresponding to $C_k$. Thus we may assume that it is monotonically decreasing over the search interval and $\frac{\partial g(\lambda, \mathbf{P})}{\partial P_k(n)}$ may be approximated by the hyperbolic model
\[
\frac{\partial g(\lambda, \mathbf{P})}{\partial P_k(n)} \approx \frac{a}{P_k(n) + C} + C_1.
\]

Fig. 1 illustrates the hyperbolic modeling of the partial derivative, where the solid blue line represents the real values of a partial derivative, while the red dotted line represents a hyperbolic model around zero.

If we detect that the partial derivative is not monotonically decreasing we may use the model
\[
\frac{\partial g(\lambda, \mathbf{P})}{\partial P_k(n)} \approx \frac{a}{P_k(n) + C} - \frac{d}{P_k(n) + D} + C_1
\]
where $a, C, d, D, C_1$ are constant parameters.

B. Root Finding Algorithm

This paragraph describes an algorithm that is able to find the root of (7) using only 4 realizations of the partial derivative. Thus the complexity of the ISB is reduced by a factor of $G/4$, where $G$ is the total number of grids used in the exhaustive line search algorithm over the interval $[0, P_{\text{max}}(n)]$.

1) First compute $\frac{\partial g(\lambda, \mathbf{P})}{\partial P_k(n)}$ at 4 equidistant points $p_1, p_2, p_3, p_4$ in the interval $[0, 0.5P_{\text{max}}(n)]$ using the exact formula 8. The values of $\frac{\partial g(\lambda, \mathbf{P})}{\partial P_k(n)}$ at these points are denoted by $A_1, A_2, A_3, A_4$.
2) If the 4 realizations of the partial derivative prove to be monotonic, approximate $\frac{\partial g(\lambda, \mathbf{P})}{\partial P_k(n)}$ by model (10).

![Fig. 1. Gradient given by exhaustive search vs its approximation around zero](Image)

- If $A_1$ and $A_3$ are both positive use the points $p_1, p_3, p_4$ to find $a, C,$ and $C_1$.
- If $A_1$ and $A_3$ are both negative or of different signs, use the points $p_1, p_2, p_3$ to find $a, C,$ and $C_1$.
- Find points $P_k^c(n)$ for which the model is equal to zero.

3) If the 4 realizations are not monotonic approximate $\frac{\partial g(\lambda, \mathbf{P})}{\partial P_k(n)}$ by model (11).
   - Use the points $p_1, p_2, p_3, p_4$ to find $a, C, d,$ and $C_1$.
   - Find the roots $P_k^c(n)$ for which the model is equal to zero.
   - For model (11) 2 roots are found. Choose the root that maximizes the objective function.

4) If $a$ is negative, the maximum is either 0 or $P_{\text{max}}(n)$.
5) If $a$ is positive, the maximum is $P_k(n) = P_k^c(n)$
6) If $a$ is positive, to improve the solution we may revisit the modeling around $P_k(n) = P_k^c(n)$.
   - Compute $\frac{\partial g(\lambda, \mathbf{P})}{\partial P_k(n)} = A_2$ at $P_k(n) = P_k^c(n)$.
   - If $P_k^c(n) > p_4, p_1 = P_k^c(n)$ and $A_1 = A_2$
   - If $P_k^c(n) < p_4, p_4 = P_k^c(n)$ and $A_4 = A_2$
   - Go to 2.

C. Complexity

The main advantage of this method with respect to existing ISB algorithm is its low complexity. The complexity of the exhaustive line search method used in conventional ISB is in the order of $O(GK^2)$, where for each user, at each possible power allocation for this user, we have to calculate the capacity for all the users ($K$ times). With $G$ is equal to the total number of possible power allocations. The root finding algorithm described above requires for each user, the computation of the
gradient ($K$ times operation), for at least 4 different power allocations, resulting in a complexity of order $O(4K^2)$. Thus the gain in complexity is of the order $G/4$.

IV. Optimization Improvement

In the previous section we presented an algorithm to reduce the overall complexity of ISB. However, the described algorithm does not enhance the outcome of the optimization itself. This section proposes two techniques to improve the optimization: The additional starting point (ASP) and the successive optimization.

A. ASP

The ISB algorithm is only guaranteed to reach a local optimum. The objective function (4) is known to have many local optima. Due to its iterative structure (iteration across the users), the result of the algorithm depends on the initial point and there is a high chance that the ISB might fall on a local optimal and not on a global one. One technique to overcome this difficulty is to use a multi start point technique. In this technique the optimization algorithm is run many times. Every time, a different initial guess is used, thus increasing the chance to find the global optimum or at least to reach a solution closer to this global optimum.

As described in section III-B, the ISB optimization is decoupled over tones. So instead of repeating the entire algorithm several times with different initial points, we propose to start with a different initial guess at each tone. Thus we can apply a multi start points procedure within a single realization of the algorithm. The use of different initial points at the different tones results in a perturbed power allocation over the tones. However combined techniques of successive optimization and multi start points can provide a smoothening effect and improve the optimization furthermore.

B. Successive Optimization

The reason behind the time sharing property described in [5] is the strong correlation between adjacent sub-channels. This correlation holds for both crosstalk and direct channels where both empirical and practical channel models show that adjacent tones have a very similar channel gain. Due to this fact we can conclude that the optimization problem at a tone $n + 1$ is usually very close to optimization at tone $n$. Thus the optimum value $P(n) = [P_1(n), P_2(n)...P_K(n)]$ that represents the power allocation at tone $n$ where $P_i(n)$ is the power allocated to user $i$ is very close to $P(n + 1)$ the optimal solution at $n + 1$.

In [10] we used this observation to propose an algorithm based on successive optimization: for each tone $n + 1$ we start an optimization based on Newton-Raphson with the result found at the previous tone $P(n)$. This simple procedure was proposed to speed up the convergence of existing algorithms as it reduces significantly the number of iterations needed for the inner loop. The same operation may be used with line search (ISB) instead of Newton-Raphson.

The drawback of successive optimization is that it may fall in the region of a local optimum for tone $n$. And due to the successive iteration it may get stuck in this region for several tones before reaching a better region.

1) Forward Successive Optimization: With these considerations in mind, successive optimization is used to smoothen the result of the ASP technique. At each tone $n + 1$ the optimization is done twice. The first optimization is started with an initial guess $P(n)$ (successive optimization). The second optimization is started with an initial guess $P(n) + E(n + 1)$ (which explains the ASP name). $E(n)$ is a random vector of the same dimension as $P(n)$ and which varies with $n$. The objective of vector $E(n)$ is to modify the initial guess $P(n)$ so that the optimization start from another region. This avoids the optimization from falling into the regions of a poor local optimum over large period of tones. At the end the outcome of the two optimizations is compared and the best one is kept.

2) Reverse Successive Optimization: The optimization proposed earlier is a forward successive optimization, where we start at tone 1 and we end at tone $N$. One way to improve the result further is to perform a reverse or backward successive optimization. This one is performed after the forward successive optimization is finished. The reverse one starts at tone $N - 1$ and moves backward to tone 1. At each tone $n$ the optimization is performed with an initial guess $P(n + 1)$. The outcome is compared to the previous result (obtained by the combined "ASP"/"Forward Successive Optimization" and the best power allocation is kept.

3) Final Procedure: Overall, the ISB algorithm is implemented 3 times for each tone. One in the forward successive mode, initialized with $P(n - 1)$; one in the ASP technique initialized with $P(n - 1) + E(n)$; and one in the reverse successive mode, initialized with $P(n + 1)$. The best solution is kept in the end.

V. Results

In this section we report the numerical results obtained for the different proposed algorithms. The direct channel gain is considered to be the attenuation loss caused by the twisted pairs, while the crosstalk channels are modeled by the 1% worst case FEXT formula [9]. The spacing between the tones is 4.3 kHz. The background noise is supposed to have a PSD of -140 dBm/Hz over all tones and for all users. A system of 7 interfering users is considered, where the users have the following distances from the CO/ONU:

<table>
<thead>
<tr>
<th>Users</th>
<th>user 1</th>
<th>user 2</th>
<th>user 3</th>
<th>user 4</th>
<th>user 5</th>
<th>user 6</th>
<th>user 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distances</td>
<td>4.8 km</td>
<td>400 km</td>
<td>3 km</td>
<td>2.4 km</td>
<td>4.5 km</td>
<td>3.8 km</td>
<td>2.3 km</td>
</tr>
</tbody>
</table>

The simulation is performed for 1024 tones, with tone zero corresponding to 258.75 kHz. A PSD mask of -30dBm/Hz is imposed over all tones. The total power allowed per user is 20 dBm. An SNR gap of 9.8 dB is considered, as well as a coding gain of 3.8 dB, and an SNR margin of 6 dB.

Table I compares the different versions of the ISB algorithm. ISB-LS refers to the original ISB algorithm with line search.
as the interior loop. ISB-LS with SO represents the original ISB algorithm with successive optimization. From the table we can see that the successive optimization speeds up the convergence at the expense of bit-rate. As explained in section IV-B, successive optimization increases the chance of staying in a local optimum region for a large range of tones. The enhanced ISB algorithm reduces considerably the execution time to less than a minute, and at the same time increases the bit-rate.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>ISB-LS</th>
<th>ISB-LS with SO</th>
<th>Enhanced Fast ISB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sum-rate</td>
<td>21877</td>
<td>21979</td>
<td>22340</td>
</tr>
<tr>
<td>Execution time</td>
<td>14.7 min</td>
<td>3.65 min</td>
<td>40 sec</td>
</tr>
</tbody>
</table>

TABLE I
COMPARISON OF THE DIFFERENT ISB ALGORITHMS

Fig. 2 plots the sum capacity over the tones for the three stages of the enhanced ISB algorithm. As expected the ASP algorithm has several ups and downs for the different tones. The forward successive optimization (FSO) algorithm manages to smoothen the capacity and to improve the optimization. However we can see, in the example given, that the FSO get stuck in local optimums for the tones $136-284$. It only reaches a better optimum at the tone $n=284$. The advantage of using the reverse successive optimization is clear. When performing the RSO the optimization at tones $136-284$ can be improved from the use of the solution at $n=284$ as starting point.

Fig. 2. Total sum capacity for the different stages of the enhanced ISB algorithm

VI. CONCLUSION

In this paper we have shown a method based on a gradient approximation to reduce the complexity of the ISB algorithm. We have also shown how to improve the performance with multi start points techniques. Finally, it was shown that a further performance gain can be obtained by taking advantage of the similarity of the channel gains between two adjacent tones using successive optimizations (both forward and reverse). These techniques allow the ISB algorithm to avoid local optimums on the different tones, which enhances the overall optimization.

ACKNOWLEDGMENT

This work was supported by NEWCOM++

REFERENCES