APPLICATION OF GRADIENT ALGORITHM FOR OPTIMIZING POWER ALLOCATION IN DSL SYSTEMS

Ali Kalakech, Jérôme Louveaux, Luc Vandendorpe

Communications and Remote Sensing Laboratory, Louvain School of Engineering, UCL
Place du Levant 2, B-1348 Louvain-la-Neuve, Belgium

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 its non-convexity, researchers had to rely on heuristic, or exhaustive research methods. In our approach we make use of the similarities of the channel gain between 2 adjacent tones to overcome the non-convexity limitation of gradient methods. This allows the proposition of an efficient and a fast routine that manages to speed up the convergence of existent state of the art algorithms.

Index Terms— DSL, DSM, Optimization, gradient

1. 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 powers 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 that was proposed 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 and independently on all modems until a constant level of powers 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. Band preference algorithm (BP) [4] is a modified version of the IWF, BP utilizes power-scaling factors in the IWF algorithm to control the bit loading process, a large scaling factor given to a particular tone would correspond to a smaller number of bit being loaded to that particular tone, and vice versa (BP could be used to protect weak users).

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) [5] 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 decoupled over the different sub channels. In order to find the optimal solution at each sub channel, OSB resorts to an exhaustive search over all the different possible power allocations which makes it too complex for practical implementation. In order to reduce the complexity of OSB, the ISB algorithm ([6, 7]) replaces the exhaustive search over all the possible power allocations by a simple line search algorithm over individual users’ powers, however this procedure is not globally optimum. Papers [8, 9] propose to solve the power allocation problem using an iterative convex approximation approach, in this approach the objective function is approximated by a convex function which allows the use of convex optimization.

In [10], we used the fact that the gain of two adjacent sub channels are close to each others to propose an optimization technique based on successive optimization (SO). This technique was able to lower considerably the execution time of the ISB algorithm. In this paper we extend the use of successive optimization to propose two additional algorithms for DSL optimization: a Newton-Raphson based algorithm (NR) and a gradient based algorithm. These two algorithms are able to update the power of all users at the same time, unlike the ISB algorithm that updates the power of only one user at each iteration. Further more the low complexity nature of the gradient algorithm will reduce the overall complexity of the optimization. This paper is decomposed as the following: Section 2 formulates the optimization problem and gives a review to different optimization algorithms. In section 3, two algorithms are proposed: one based on Newton-Raphson and the other on steepest ascent. Finally the numerical results are reported for different cases in section 4.

2. SYSTEM MODEL AND ALGORITHMS REVIEW

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

2.1. 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

\[
SNR_i(n) = \frac{|H_{ii}(n)|^2 P_i(n)}{\sigma_i^2(n) + \sum_{n \neq i} |H_{ii}(n)|^2 P_i(n)}.
\]

Where \( H_{ii}(n) \) is the direct channel gain of user \( i \) at tone \( n \). \( H_{il}(n) \) 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:

\[
\max_{\mathbf{P}} \sum_{i=1}^{K} \omega_i R_i
\]

Subject to

\[
\sum_{n=1}^{N} P_i(n) \leq P_t,
\]

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

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

2.2. 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 [5, 7] try to solve this problem globally by introducing the dual function to relax the power constraints. The Lagrange function of problem (3) is given by:

\[
g(\lambda, \mathbf{P}) = \sum_{i=1}^{K} \omega_i R_i - \lambda_i \left( \sum_{n=1}^{N} P_i(n) - P_t \right)
\]

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

\[
f(\lambda) = \max_{\mathbf{P}} g(\mathbf{P}).
\]

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 \( g(\mathbf{P}) \) with respect to \( \mathbf{P} \). A simple sub-gradient algorithm was proposed in [6, 7] for finding \( \lambda \) in the outer loop. The search for \( \lambda \) was improved for better convergence in [11]. Examining \( g(\mathbf{P}) \) shows that for fixed \( \lambda \) there is no coupling between the tones \( n \) as \( \sum_{n}^{K} \lambda_i P_t \) 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, ie for the inner loop optimization, 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 \( P_i(n) \) of the remaining user to maximize \( g(\mathbf{P}) \). This procedure is repeated iteratively over all users till a constant power allocation is reached.

2.3. Successive Optimization

In DSL systems, there is a strong correlation between adjacent subchannels. 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 very close to optimization at tone \( n \). Thus the optimum value \( \mathbf{P}(n) = [P_1(n), P_2(n)\ldots P_K(n)] \) that represents the power allocation found at tone \( n \) (with \( P_i(n) \) is the power allocated to user \( i \)) is very close to \( \mathbf{P}(n+1) \) the optimal solution at \( n + 1 \).

In [10], we proposed the use of successive optimization (SO) as a method of enhancement of the ISB algorithm. To use the SO: For each tone \( n + 1 \) we start the line search over the power with with the result found at the previous tone \( P_t \). This simple procedure was able to speed up the convergence of ISB as it reduces significantly the number of iterations needed for the inner loop. The results in [10] shows the validity of the successive optimization approach, where 2 adjacent tones’ optimums were found to lie at proximity of each other for most of the tones.

2.4. Gradient Algorithm

In gradient algorithms, and in order to maximize the objective function around the point \( \mathbf{P}(n) \), one should take steps proportional and of the same direction as the gradient \( \nabla g(\mathbf{P}(n)) \):

\[
\mathbf{P}(i+1) = \mathbf{P}(i) + \alpha \nabla g(\mathbf{P}(i))
\]

Where \( i \) is the iteration index, and \( \alpha \) is a positive small step size. In this case (\( \alpha > 0 \)), the gradient is called gradient ascent or steepest ascent. The steepest ascent procedure guarantees the convergence toward the nearest local optimum that lies in the proximity of the initial point \( \mathbf{P}(0) \). In the case of convex optimization, the steepest ascent (SA) algorithm will almost certainly find the global maximum, however if the optimization is a non convex one, SA may converge into a
poor local optimum. The next section shows that when coupled with SO, the gradient algorithm can be used for the optimization of DSL systems, even if the optimization is suffering from multiple local optima.

3. GRADIENT TYPE ALGORITHMS

As it was previously shown, the optimum power allocations at two adjacent tones (P(n) and P(n + 1)) are at vicinity of each others. The vector P(n + 1) is considered to be at least a local maximizer of the objective function g(P) at the tone n + 1. This means that g(P) will be typically strictly concave in the neighborhood of P(n + 1). Since P(n) lies in this neighborhood, initializing the optimization with P(n) put us in good situation to use gradient type algorithms. This helps reducing the computation time considerably. In this section, first we will describe an algorithm based on Newton-Raphson, then we will implement a steepest ascent gradient algorithm.

3.1. Newton Raphson

Newton Raphson method approximate the objective function around P(n) by a quadratic function. So according to this method, the correction applied to vector P(n) at iteration i is given by

$$P^{(i)}(n) = P^{(i-1)}(n) - C^{-1} \nabla g[P^{(i-1)}(n)]$$  

(8)

where C is the Hessian of the Lagrange function g, evaluated for P^{(i-1)}(n) and \nabla g[P^{(i-1)}(n)] is the gradient at the same value. It may happen that the initial guess proposed is not close enough to the optimum value. This can be detected when the Hessian is non-negative definite. In such a case the NR method can no longer be used. Therefore we resort to gradient method for a few iterations, until the Hessian becomes negative definite. The Hessian inversion required for the NR method can be seen as draw back for this procedure. That's why we propose next a quasi optimal steepest ascent algorithm that requires no matrix inversion.

3.2. Quasi Optimal Steepest Ascent

For optimal steepest ascent the correction step \alpha corresponds to \alpha_m, which is given by:

$$\alpha_m = \max_{\alpha} \ g(P(n) + \alpha \nabla g[P(n)])$$  

(9)

Thus at each iteration, the optimal SA searches for the step size \alpha_m that maximizes the objective function in the direction of the gradient. As in (3), problem (9) is a non convex problem. One way to find \alpha_m is to find all the critical points and then testing them for optimality. To find all the critical points, one should find all the points \alpha that satisfy the equation:

$$\frac{\partial}{\partial \alpha} g(P(n) + \alpha \nabla g[P(n)]) = 0$$  

(10)

Defining the elements of \nabla g[P(n)] by:

$$G_i(n) = \frac{\partial}{\partial P_i} g[P(n)]$$

Equation (10) can be rewritten as:

$$\sum_{i}^{K} \frac{\omega_i^{(n)}}{(\eta_i(n) + \alpha)(\mu_i(n) + \alpha)} + \sum_{i}^{K} \lambda_i G_i(n) = 0$$  

(11)

with the terms:

$$\omega_i^{(n)} = \omega_i^{(n)} \frac{b_i(n)c_i(n) - a_i(n)d_i(n)}{d_i(n)(\sigma_i(n) + d_i(n))}$$  

(12)

$$\eta_i(n) = \frac{a_i(n) + b_i(n)}{c_i(n) + d_i(n)}$$  

(13)

$$\mu_i(n) = \frac{b_i(n)}{d_i(n)}$$  

(14)

$$a_i(n) = |H_{ii}^{(n)}(P_i(n)$$  

(15)

$$b_i(n) = \sum_{l \neq i} |H_{ii}^{(n)}(P_l(n) + \sigma_l(n)$$  

(16)

$$c_i(n) = |H_{ii}^{(n)}(G_i(n)$$  

(17)

$$d_i(n) = \sum_{l \neq i} |H_{ii}^{(n)}(G_l(n)$$  

(18)

Fig.1 shows a typical plot of the LHS of equation (11) in function of \alpha. The different asymptotes shown in Fig.1 corresponds to \alpha = -\eta_i(n) or \alpha = -\mu_i(n). Since \alpha must be strictly positive, we are only concerned about negative \eta_i and \mu_i.

Let A_1=\min(-\mu_i(n),-\eta_i(n)) for all negative \mu_i(n) and \eta_i(n). Since the channel gains are strictly positive, it can be easily shown that for \alpha \geq A_1 at least one element of the vector P^{(i)}(n) = P^{(i-1)}(n) + \alpha \nabla g[P^{(i-1)}(n)] must be negative. In practice these negative power components correspond to the lines being not active at the given tones. Thus if the optimum value requires transmitting powers on all lines, \alpha_m should be typically between 0 and A_1. To find \alpha_m between 0 and A_1 a hyperbola was used to approximate (11) between these two points.

The model \gamma = \frac{al}{d + A_1} + C_{st} is found to give good results. Parameters \gamma and C_{st} can be calculated using two values of equation (11) (In the simulation equation (11) is evaluated around 0 and A_1/2). The approximate value of \alpha_m will be given by \alpha_m = -al/C_{st} - A_1. Another hyperbolic approximation near \alpha_m may be done to improve the solution.
An optimum value may exist for $\alpha_m$ larger than $A_1$. In this case the negative resultant power must be set to null which results in a new optimization with a lesser number of active lines. To check for optimality for each positive asymptotes replace the negative power with zeros and get the optimal value for the remaining active lines.

### 4. 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, the spacing between the tones is 4.3 kHz.

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 m</td>
<td>5 km</td>
<td>2.4 km</td>
<td>3.5 m</td>
<td>3.8 km</td>
<td>2.3 km</td>
</tr>
</tbody>
</table>

The simulation is done 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.

<table>
<thead>
<tr>
<th>Number of users</th>
<th>3 users</th>
<th>5 users</th>
<th>7 users</th>
</tr>
</thead>
<tbody>
<tr>
<td>SA</td>
<td>21542</td>
<td>24007</td>
<td>26944</td>
</tr>
<tr>
<td>NR</td>
<td>20562</td>
<td>23885</td>
<td>26896</td>
</tr>
<tr>
<td>ISB</td>
<td>21516</td>
<td>24029</td>
<td>26833</td>
</tr>
</tbody>
</table>

**Table 1.** Sum capacity obtained for different algorithms in bps/Hz

The simulations are done in the presence of the first 3 users only, then for the first 5 users, and finally for all 7 users. Table 1 gives the sum capacities for the three simulations using different optimization algorithms. Table 2 gives the average execution of internal loops required for each one of these simulations.

<table>
<thead>
<tr>
<th>Number of users</th>
<th>3 users</th>
<th>5 users</th>
<th>7 users</th>
</tr>
</thead>
<tbody>
<tr>
<td>SA</td>
<td>1.4</td>
<td>3.7</td>
<td>7.6</td>
</tr>
<tr>
<td>NR</td>
<td>1.8</td>
<td>4.3</td>
<td>11</td>
</tr>
<tr>
<td>ISB</td>
<td>42</td>
<td>120</td>
<td>240</td>
</tr>
</tbody>
</table>

**Table 2.** Execution time for different algorithms in seconds

The ability to test for different values of $\alpha$ in the quasi-optimal steepest ascent guarantee that at each step the algorithm provides a near global optimal solution in the gradient direction even when it initializes at a point far from optimal or when we have multiple optimums. Thats why in the two scenarios explained above the SA algorithm provided near optimal results with relatively smaller complexity.

In the Newton-Raphson case the algorithm tends to follow the local optimum that lies at vicinity of the initial guess. The argument that "adjacent tones' optimums are close to each others and lie in the same region" works reasonably well, but over a large number of tones the global optimum may slip away from the initial optimal region while the NR algorithm is still stuck in it with local optimums.

### 5. CONCLUSION

In this paper we have shown how to take advantage of the similarity of the channel gains between two adjacent tones by proposing a simple technique based on successive optimization. This techniques allows the implementation of fast gradient type algorithms for the DSL spectrum optimization. It may also be used to speed up the convergence of the existent state of the art algorithms such as ISB.

### 6. REFERENCES


