Autonomous Spectrum Management (ASB) for Asynchronous DSL

Raphael Cendrillon\(^1\), Jianwei Huang\(^2\), Mung Chiang\(^2\)

\(^1\) Electrical Engineering, University of Queensland, Australia
\(^2\) Electrical Engineering, Princeton University, USA

Abstract—Dynamic spectrum management is an area of growing importance in DSL research and can dramatically increase data rate and service reach. Unfortunately most DSM algorithms operate under the assumption of perfect synchronization between the DSL modems within a network, an assumption that is almost never satisfied in practice. Lack of synchronization means that modems see crosstalk from one-another not only on the corresponding tone, but on neighboring tones as well. This inter-carrier interference increases the difficulty of optimizing the transmit spectra, coupling the spectrum management problem among tones. In this paper we propose two power allocation algorithms for the asynchronous DSL network, based on the Asynchronous Spectrum Balancing (ASB) algorithm. We name them Asynchronous ASB (A-ASB) algorithms, which include Greedy Power Shuffle (GPS) algorithm and High SINR Approximation (HSA) algorithm. Both A-ASB algorithms are completely autonomous and have lower complexity than the current state-of-the-art algorithms proposed in [1]. Convergence of the algorithms is shown under reasonable conditions that are satisfied in typical DSL channels. The GPS version of the A-ASB algorithm achieves significant better performance than the ASB algorithm.

I. INTRODUCTION

Digital Subscriber Line (DSL) technologies transform traditional voice-band copper channels into high bandwidth data pipes, which are capable of delivering data rates up to several Mbps per twisted-pair over a distance of 10 kft. The major obstacle for performance improvement in modern DSL systems (e.g., ADSL and VDSL) is the crosstalk, which is the interference generated between different lines in the same binder. In the case of perfect synchronization between the different Discrete MultiTone (DMT) transmission blocks, the crosstalk experienced by a line on a certain tone is due to the transmissions of other lines on the same tone. In practice, however, perfect DMT synchronization cannot be achieved due to differences in channel propagation delays. In that case, orthogonality among tones are destroyed and inter-carrier-interference (ICI) leads to more serious crosstalks. In both the synchronous and asynchronous cases, dynamic spectrum management (DSM) can significantly improve data rates over the current practice of static spectrum management that mandates spectrum mask or flat power backoff across all frequencies (i.e., tones).

Most of the recently proposed DSM algorithms focus on the synchronous transmission case, including the Iterative Waterfilling (IW) algorithm [2], the Optimal Spectrum Balancing (OSB) algorithm [3], the Iterative Spectrum Balancing (ISB) algorithm [4], and the Autonomous Spectrum Balancing (ASB) algorithm [5]. Among all the algorithms, the ASB is completely autonomous with the lowest computational complexity (similar as IW), but achieves a performance much better than IW and close to the centralized algorithms, OSB and ISB. All these algorithms utilize the dual-based decomposition technique by relaxing modem’s individual power constraints and making the spectrum balancing problem separable across tones. As a result, they are not directly applicable in the asynchronous transmission case, since dual-based relaxation here will not make the problem separable due to the additional coupling across tones caused by ICI. Moreover, the duality gap of the spectrum sharing problem is no longer asymptotic zero as argued in [3], since the time-sharing (frequency-sharing) property no longer holds.

This paper develops, analyzes, and simulates two new algorithms for power allocation (or, equivalently, bit loading) in the asynchronous transmission case, based on the ASB originally proposed in the synchronous case [5]. To differentiate from [5], we call the two algorithms here as Asynchronous Autonomous Spectrum Balancing (A-ASB) algorithms. Both versions of the A-ASB algorithms are autonomous (distributed across the users with no explicit information exchange) with low complexity, while provably convergent (under certain sufficient conditions on channel gains) and achieve better performance than the algorithms without considering ICI, thus overcoming the bottlenecks of the state-of-art algorithms proposed in [1].

In [1], the author proposed two centralized greedy power allocation algorithms, bit-subtracting and bit-adding algorithms, which start from the power spectrum density (PSD) obtained with the ISB algorithm and search for local optimal solutions in the neighborhood by taking ICI into account. Due to the centralized nature of these algorithms, they are computational expensive in the case of large numbers of users and tones. Moreover, centralized computation is often unrealistic in DSL networks. First, there exist regulatory requirements on “unbundling” service, i.e., incumbent service providers must rent certain lines to their competitors. This makes it very costly to perform a centralized optimization. Also, many lines in the same binder terminate on different quad cards in the DSL Access Multiplexer because customers in the same neighborhood sign up at different times, which makes it impossible to have central coordination even if one can tolerate the cost issues.

The two versions of the A-ASB algorithms introduced in this paper, also called the Greedy Power Shuffle (GPS) algorithm and High SINR Approximation (HSA) algorithm, reduce the complexity from those in [1], and achieve significant better performance than the ASB algorithm in [5]. The basic idea here, similar as in [5], is to use the concept of reference line...
to mimic a “typical” victim line. By setting the power spectrum level to protect the reference line, a good balance between local and global maximization can be achieved. Compared with [5],
determining the optimal PSDs are much more complex here
due to the presence of ICI. In particular, the GPS algorithm
searches for the local optimal PSD solution of each user jointly
across all tones without using the dual decomposition as in [5].
In the HSA algorithm where the dual-decomposition becomes
possible after approximations, we find an analytical solution
that is a generalization of the frequency selective water-filling

Denote $K$ as the number of tones and $N$ the number of
users, then both GPS and HSA algorithms enjoy a linear
complexity in $N$, and the HSA enjoys a linear complexity
in $K$ due to dual-decompositions. We prove the convergence
of both algorithms, under sequential and parallel updates. The
comparison of our algorithms with the ones in [1] are listed in
Table I, where “RC” in the second line represents for “Reduced
Complexity”.

### II. SYSTEM MODEL

We consider a DSL network with $\mathcal{N} \{1,\ldots,N\}$ users (i.e.,
lines, modems) and $\mathcal{K} \{\cdots,K\}$ tones (i.e., frequencies,
carriers). Denote $x_k^n$, $y_k^n$ and $n_k^n$ as the transmitted signal,
received signal and the background noise of user $n$ on tone $k$.
Also denote $s_k^n \equiv \mathbb{E}\{|x_k^n|^2\}$ as the transmit power spectrum
density (PSD). The received PSD of user $n$ on tone $k$ is

$$
\mathbb{E}\{|y_k^n|^2\} = |h_k^{n,m}|^2 s_k^n + \sum_{m \neq n} \sum_{j=1}^K \gamma(k-j)|h_j^{n,m}|^2 s_j^m + \mathbb{E}\{|n_k^n|^2\},
$$

where $\mathbb{E}\{\cdot\}$ means expected value. Here $h_k^{n,m}$ is the channel
gain from user $m$’s transmitter to user $n$’s receiver on tone $k$.
$\gamma(j)$ is the ICI coefficients estimated in the worst case [1],

$$
\gamma(j) = \begin{cases} 
2, & j = 0 \\
\frac{2}{K^2 \sin^2\left(\frac{\pi j}{K}\right)}, & -\frac{K}{2} \leq j < \frac{K}{2}, j \neq 0 
\end{cases}
$$

and has the symmetric and circular properties, i.e., $\gamma(-j) = \gamma(j) = \gamma(K-j)$. A numerical example is shown in Figure 1
for $K = 256$.

Assume that each user treats interference from other users
as noise. When the number of interfering users is large,
the interference can be well approximated by a Gaussian
distribution. Under this assumption the achievable bit loading
of user $n$ on tone $k$ is

$$
b_k^n \triangleq \log\left(1 + \frac{1}{\Gamma} \sum_{m \neq n} \left(\sum_{j=1}^K \gamma(k-j)\alpha_j^{n,m} s_j^m + \sigma_j^n\right)\right)
$$

where $\alpha_j^{n,m} = |h_j^{n,m}|^2 / |h_j^{n,n}|^2$ is the normalized crosstalk
channel gain, and $\sigma_j^n = \mathbb{E}\{|n_j^n|^2\} / |h_j^{n,n}|^2$ is the normalized
noise power. Here $\Gamma$ denotes the SINR-gap to capacity, which
is a function of the desired BER, coding gain and noise margin
[6]. For notational simplicity, we absorb $\Gamma$ into the definition
of $s_k^n$. The data rate on line $n$ is thus $R^n = f_s \sum_{k \in \mathcal{K}} b_k^n$,
where $f_s$ denotes the symbol rate. Each modem is typically
subject to a total power constraint $P^n$, due to the limitations
on each modem’s analog frontend: $\sum_{k \in \mathcal{K}} s_k^n \leq P^n$.

### III. SPECTRUM MANAGEMENT PROBLEM FORMULATION

The spectrum management problem is defined as follows

$$
\max_{\{s^n, n \in \mathcal{N}\}} R^1 \quad \text{s.t.} \quad R^n \geq R^{n,\text{target}}, \forall n > 1, \quad \text{(1)}
\sum_{k \in \mathcal{K}} s_k^n \leq P^n, \forall n.
$$

Here $R^{n,\text{target}}$ the target rate constraint of user $n$, and $s^n = (s_k^n)_{k \in \mathcal{K}}$ is the PSD of user $n$. Problem (1) is a nonconvex
(due to mutual interference) and highly coupled (across lines and
tones) optimization problem. In particular, any algorithm
trying to optimally solve it inevitably requires knowing
the global information. Different than the spectrum balancing
problem considered in the synchronized case [3], [4],
the duality gap of Problem (1) cannot be asymptotically reduced
to zero even with large enough $K$, which precludes the direct
application of dual decomposition methods used in most of
the previous literatures.

The author in [1] considered a discrete bit loading version
of Problem (1) (i.e., $b_k^n$ needs to be integer for all $n$ and $k$),
and proposed two heuristic algorithms to achieve local optimal
solutions of Problem (1) in the two-user case. The proposed
algorithms have complexity of $O(N^3 K^3)$ (or $O(N^2 K^3)$
when the power series approximation converges), and is quite
computational expensive in the case of many users and many
tones.

### TABLE I

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Operation</th>
<th>Complexity</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Greedy algorithm</td>
<td>Centralized</td>
<td>$O(N^2 K^3)$</td>
<td>[1]</td>
</tr>
<tr>
<td>RC Greedy algorithm</td>
<td>Centralized</td>
<td>$O(N^3 K^3)$</td>
<td>[1]</td>
</tr>
<tr>
<td>GPS</td>
<td>Autonomous</td>
<td>$O(N K^2 \log_2(K))$</td>
<td>this paper</td>
</tr>
<tr>
<td>HSA</td>
<td>Autonomous</td>
<td>$O(N K)$</td>
<td>this paper</td>
</tr>
</tbody>
</table>

![ICI coefficients for $K = 256$](image)
We observe that at the optimal solution(s) of Problem (1), each user chooses a PSD level that leads to a good balance of maximization of his own rate and minimization of the damages he causes to the other users. Based on this insight, we introduced the concept of a “reference line”, a virtual line that represents a “typical” victim in [5]. Then instead of solving Problem (1), each user tries to maximize the achievable data rate on the reference line, subject to its own data rate and total power constraint. Define the rate of the reference line to user \( n \) as

\[
R^{n,\text{ref}} \triangleq \sum_{k \in K} \hat{b}^n_k \triangleq \sum_{k \in K} \log \left( 1 + \frac{\tilde{s}_k}{\sum_{j=1}^{K} \gamma (k - j) \tilde{\alpha}^n_j s^n_j + \tilde{\sigma}_k} \right).
\]

The coefficients \( \{ \tilde{s}_k, \tilde{\sigma}_k, \tilde{\alpha}^n_k \} \) are fixed parameters of the reference line and can be obtained from field measurements. These have been defined by the network operators in the current standards, and represent the worst-case interference seen in a typical binder. Each user \( n \) then wants to solve the following problem local to himself:

\[
\max_{s^n} R^{n,\text{ref}} \quad \text{s.t.} \quad R^n \geq R^n,\text{target}, \\
\sum_{k \in K} s^n_k \leq P^n. \tag{2}
\]

By using Lagrangian relaxation on the target rate constraint in Problem (2) with a weight coefficient (dual variable) \( w^n \), the relaxed version of Problem (2) is

\[
\max_{s^n} w^n R^n + R^{n,\text{ref}} \quad \text{s.t.} \quad \sum_{k \in K} s^n_k \leq P^n. \tag{3}
\]

The weight coefficient \( w^n \) needs to be adjusted to enforce the rate constraint.

IV. ASYNCHRONOUS ASB ALGORITHMS

Problem (3) is highly non-convex and highly coupled due to crosstalk. Different from the synchronous DSL case considered in [5], a dual-based decomposition is not applicable here since the PSD across different tones are coupled due to ICI.

We first introduce a greedy power shuffle (GPS) algorithm, where we start from the PSD level achieved by running the ASB algorithm, and then let each user \( n \) sequentially “shuffle” its PSD \( s^n = \{ s^n_k, k \in K \} \) (i.e., subtract a small amount from one tone and add it back to another tone), to reach a local optimal solution of Problem (3). However, the convergence of GPS is difficult to prove, although it is always observed in practice. Then we introduce another variation of the ASB algorithm under high SINR approximation (HSA) of the reference line, which enjoys lower computational complexity and provable convergence.

1Depending on different time period of the day, week or month, the coefficients \( \{ \tilde{s}_k, \tilde{\sigma}_k, \tilde{\alpha}^n_k \} \) could take different values. However, they are fixed for the optimization problem considered in this paper.

A. Greedy Power Shuffle (GPS) Algorithm

In the GPS algorithm, we start the algorithm by running the ASB algorithm for the synchronized DSL system in [5], which leads to a feasible initial PSD allocation for all users. In fact, the power constraints for all users are tight after this initialization, since typically the reference line is inactive in the high frequency tones, thus it is beneficial to transmit all possible remaining power in these tones in the synchronous case in terms of solving Problem (3).

After initialization, each user \( n \) tries to solve Problem (3) for a fixed \( w^n \), then adjusts \( w^n \) to make the target rate tight. Unlike the synchronized case in [5], here the objective function in Problem (3) is not separable across tones due to ICI, thus a dual-based relaxation of the power constraint will not lead to a decomposable problem structure.

The approach we take here is to let each user search for a better PSD allocation from the initial conditions, until no improvement can be found. To this end, denote the objective function of Problem (3) as \( J^n (s^n) \). Also define \( \Delta s \) as the incremental amount of power a user can change on a tone at a time. In other words, \( \Delta s \) define the granularity of the local search, which trades off performance and convergence speed.

For each user \( n \) with fixed \( w^n \), each search iteration consists of two phases: subtraction phase and addition phase. In the subtraction phase, user \( n \) reduces its PSD by \( \Delta s \) on the tone that yields the maximum increase in \( J^n (s^n) \) (or the smallest decrease if decreasing \( \Delta s \) on any tone leads a decreased objective). In the addition phase, user \( n \) increases its PSD by \( \Delta s \) on the tone that yields the maximum increase in \( J^n (s^n) \) (or smallest decrease similar as in the subtraction phase). This iteration repeats until the net change of \( J^n (s^n) \) in the last iteration (i.e., the sum of change in both phases) is zero. Note that the net change of objective function will never be negative for a single iteration, since in the addition phase a user can always add \( \Delta s \) back to the same tone chosen in the subtraction phase and achieve exact the same PSD level as in the previous iteration.

Algorithm 1 The Greedy Power Shuffle (GPS) algorithm

1: \( w \leftarrow e \).
2: \( s^n \leftarrow \left( P^n / K \right) e \) for all \( n \in N \).
3: repeat
4: for all \( n \in N \) do
5: repeat
6: \( s^n \leftarrow \text{ASB} (\tilde{\alpha}^n, \tilde{\sigma}, IN^n, w^n, n) \).
7: \( s^n \leftarrow PS (\tilde{\alpha}^n, \tilde{\sigma}, IN^n, \gamma, w^n, n) \).
8: \( w^n = [w^n - \varepsilon (R^n (s^n) - R^n,\text{target})]^{+} \).
9: until \( R^n = R^n,\text{target} \)
10: end for
11: until \( R^n = R^n,\text{target} \) for all \( n \)

The complete GPS algorithm is given in Algorithm 1. Line 1 initializes users’ weight coefficients \( w = (w^n)_{n \in N} \), and line 2 initializes all users’ PSD by allocating constant power across tones. Here \( e \) is a properly dimensioned vector with all entries equal to 1. Line 6 computes user \( n \)’s PSD by running the ASB algorithm [5], given fixed crosstalk to the reference line \( \tilde{\alpha}^n =
\((\tilde{a}_k^n)_{k \in K^n}\) noise power on the reference line \(\tilde{\sigma} = (\tilde{\sigma}_k)_{k \in K^n}\), received interference plus noise from other actual lines \(IN_n = (IN_k^n)_{k \in K^n}\) and weight coefficient \(w^n\). Line 7 updates the PSD by running the PS subroutine that further takes the ICI coefficients \(\gamma = (\gamma (k))_{k \in K^n}\) into consideration. Finally, user \(n\) updates \(w^n\) in line 8 with a fixed constant stepsize to make the target rate constraint tight.

The Power Shuffle subroutine is specified in Algorithm 2. Line 3 finds the set of tones on which a decrease of PSD is always guaranteed, mainly due to the nonconvexity of Problem (3). The convergence of the GPS algorithm in Algorithm 1 is not always tight. This is because we take \(\Delta s\) away from one tone in the subtraction phase, and put it back to one tone in the addition phase. Thus the resource is always fully utilized and no power violation will occur. This is different from the bit-addition and bit-subtraction algorithms in [1], where the power constraints are either loose or violated during the whole process of the algorithm before convergence.

Second, each user \(n\) always achieves a better objective \(J^n(s^n)\) at the end of the PS subroutine, compared with the one achieved by the ASB before the PS subroutine. This is due to the monotonic increase of \(J^n(s^n)\) during the iterations of the subroutine.

Third, the overall complexity of the GPS algorithm is \(O(N K^2 \log_2 (K))\), which is reasonable for a large number of users \(N\) and practical values of \(K\) in the current standards (e.g., \(N\) is usually between 50 and 100; \(K = 256\) for ADSL and \(K = 4096\) for VDSL). The linear complexity in \(N\) is due to the decoupling across users. During each iteration of the PS subroutine, each user needs to search through \(K\) possible tones to decide which tone to subtract (add) power from (to). And for each tone choice, we need to calculate the corresponding data rate. This requires a convolution of the PSDs in the frequency domain, which has a complexity of \(K \log_2 (K)\) if we implement it with an FFT. In comparison, the proposed bit-adding and bid-subtracting algorithms in [1] have complexity of \(O(N^3K^2)\) (or \(O(N^2K^3)\) when the power series approximation converges), thus is computational expensive for the case of many users and tones in a typical DSL network.

**Algorithm 2 Power Shuffle (PS) subroutine**

1: procedure PS\((x^n, \tilde{\sigma}, IN^n, \gamma, w^n, n)\)
2:   repeat
3:     \(K^{pos} \leftarrow \{k : s^n_k \geq \Delta s\}\).
4:     for all \(k' \in K^{pos}\) do
5:         \(s^n_k \leftarrow w^n\).
6:         \(s^n_k \leftarrow s^n_k - \Delta s\).
7:         \(\Delta J^n_k (s^n_k) \leftarrow J^n(s^n_k) - J^n(s^n_k)\).
8:     end for
9:     \(k_{op} \leftarrow \arg \max_{K^{pos}} \Delta J^n(s^n_k)\).
10:    \(s^n_{k_{op}} \leftarrow s^n_{k_{op}} - \Delta s\).
11:   for all \(k' \in K\) do
12:       \(s^n_k \leftarrow w^n\).
13:       \(s^n_k \leftarrow s^n_k + \Delta s\).
14:       \(\Delta J^n_k (s^n_k) \leftarrow J^n(s^n_k) - J^n(s^n_k)\).
15:   end for
16:   \(k_{op} \leftarrow \arg \max_{K} \Delta J^n(s^n_k)\).
17:   \(s^n_{k_{op}} \leftarrow s^n_{k_{op}} + \Delta s\).
18:   \(\Delta J^n = \Delta J^n (s^n_{k_{op}}) + \Delta J^n (k_{op})\).
19: until \(\Delta J^n = 0\)
20: return \(s^n\)
21: end procedure

It is clear that the following is true:

**Proposition 1:** The PS subroutine always converges.

**Proof:** Since the value of \(J^n(s^n)\) increases in each iteration and is upper-bounded, it must converge. 

The convergence of the GPS algorithm in Algorithm 1 is not always guaranteed, mainly due to the nonconvexity of Problem 3 and the fact that the PS subroutine can only reach a local optimal solution (in a predefined granularity determined by \(\Delta s\)). In our simulation, however, the GPS algorithm always converges.

There are several interesting properties of the GPS algorithm. First, At the end of each iteration of the PS subroutine, the power constraint of user \(n\) is always tight. This is because we take \(\Delta s\) away from one tone in the subtraction phase, and put it back to one tone in the addition phase. Thus the resource is always fully utilized and no power violation will occur. This is different from the bit-addition and bit-subtraction algorithms in [1], where the power constraints are either loose or violated during the whole process of the algorithm before convergence.

Due to the high coupling induced by ICI, it is very difficult to find the global optimal solution of Problem (3). In the GPS algorithm, we resort to finding a locally optimal solution with finite granularity, which involves quite some computation. However, due to the power constraints, the overall complexity of the algorithm is small. To further reduce the computation complexity, and to gain more insight into the solution structure, we assume that the reference line operates in the high SINR regime whenever it is active, i.e., if \(s_k > 0\), then \(s_k \gg \tilde{\sigma}_k \gg \sum_{j=1}^{K} \gamma (k-j) \tilde{a}_j^n s_j^n\) for any feasible \(s^n\), \(n \in N\). This assumption is motivated by our observations of the optimal solutions in DSL type of interference channels. It means that the reference PSD is much larger than the reference noise, which is in turn much larger than the interference from user \(n\). Then the reference line’s achievable rate on tone \(k\) is

\[
\log \left( 1 + \frac{\hat{s}_k}{\sum_{j=1}^{K} \gamma (k-j) \tilde{a}_j^n s_j^n + \tilde{\sigma}_k} \right) \\
\approx \log \left( \frac{\hat{s}_k}{\tilde{\sigma}_k} \right) 1_{\{\hat{s}_k > 0\}} - \frac{\sum_j \gamma (k-j) \tilde{a}_j^n s_j^n}{\tilde{\sigma}_k} 1_{\{\hat{s}_k > 0\}}
\]

where \(1_{\{\cdot\}}\) is the indicator function, i.e., equals to 1 if \(\hat{s}_k > 0\).

If we further relax user \(n\)’s total power constraint by a dual variable \(\lambda^n\), then the objective function in Problem (3) can be approximated by

\[
\log \left( \frac{\hat{s}_k}{\tilde{\sigma}_k} \right) 1_{\{\hat{s}_k > 0\}} - \frac{\sum_j \gamma (k-j) \tilde{a}_j^n s_j^n}{\tilde{\sigma}_k} 1_{\{\hat{s}_k > 0\}}
\]

\[
\sum_k \left( \log \left( \frac{\hat{s}_k}{\tilde{\sigma}_k} \right) 1_{\{\hat{s}_k > 0\}} - \frac{\gamma (k-j) \tilde{a}_j^n s_j^n}{\tilde{\sigma}_k} 1_{\{\hat{s}_k > 0\}} \right)
\]

\[
\sum_k \left( \log \left( \frac{\hat{s}_k}{\tilde{\sigma}_k} \right) 1_{\{\hat{s}_k > 0\}} - \frac{\sum_j \gamma (k-j) \tilde{a}_j^n s_j^n}{\tilde{\sigma}_k} 1_{\{\hat{s}_k > 0\}} \right)
\]

\[
\sum_k \left( \log \left( \frac{\hat{s}_k}{\tilde{\sigma}_k} \right) 1_{\{\hat{s}_k > 0\}} - \frac{\sum_j \gamma (k-j) \tilde{a}_j^n s_j^n}{\tilde{\sigma}_k} 1_{\{\hat{s}_k > 0\}} \right)
\]

\[
\sum_k \left( \log \left( \frac{\hat{s}_k}{\tilde{\sigma}_k} \right) 1_{\{\hat{s}_k > 0\}} - \frac{\sum_j \gamma (k-j) \tilde{a}_j^n s_j^n}{\tilde{\sigma}_k} 1_{\{\hat{s}_k > 0\}} \right)
\]

\[
\sum_k \left( \log \left( \frac{\hat{s}_k}{\tilde{\sigma}_k} \right) 1_{\{\hat{s}_k > 0\}} - \frac{\sum_j \gamma (k-j) \tilde{a}_j^n s_j^n}{\tilde{\sigma}_k} 1_{\{\hat{s}_k > 0\}} \right)
\]
The corresponding optimal PSD that maximizes \( J^n (s^n) \) is

\[
\begin{align*}
\mathbf{s}^n_{\text{HSA}} (w^n) &= \left[ \frac{w^n}{\lambda^n + \hat{\alpha}^n_k j \sum_j \gamma_k j - k \alpha_j n, m s_j m} - \hat{\sigma}^k \right]_{\geq 0},
\end{align*}
\]

where \( \lambda^n \) is chosen to make the total power constraint tight, \( \sum_k s^n_{\text{HSA}} (w^n) = P^n \). Here we emphasize the dependence of the solution on \( w^n \). This is a water-filling type of solution and is intuitively satisfying: the optimal PSD should be smaller when the power constraint is tighter (i.e., \( \lambda^n \) is larger), or the interference coefficient to the reference line \( (\hat{\alpha}^n_k) \) is higher, or noise level on the reference line \( (\hat{\sigma}^k) \) is smaller, or there is more interference plus noise \( \left( \sum_{m \neq n} \sum_j \gamma_k j - k \alpha_j n, m s_j m + \hat{\sigma}^k \right) \) on the current tone. This is a generalization of the frequency selective waterfilling proposed in [5], where the \( s^n_k \) is determined only by parameters and variables on tone \( n \). The complete HSA algorithm is given in Algorithm 3. Note that the determination of user \( n \)’s PSD in line 6 according to (4) involves an iterative search of the optimal \( \lambda^n \). The computational complexity of the HSA algorithm is \( O(NK) \), where \( N \) is due to iterations through the users, and \( K \) is due to the calculation of the optimal PSD on each tone according to (4).

Algorithm 3 High SINR Approximation (HSA) algorithm
1: \( w^n \leftarrow e \).
2: \( s^n \leftarrow \left( P^n / K \right) e \) for all \( n \in N \).
3: repeat
4: for all \( n \in N \) do
5: repeat
6: \( s^n \leftarrow s^n_{\text{HSA}} (w^n) \)
7: \( w^n = \left( w^n - \varepsilon \left( R^n (s^n) - R^n_{\text{target}} \right) \right)^+ \).
8: until \( R^n = R^n_{\text{target}} \)
9: end for
10: until \( R^n = R^n_{\text{target}} \) for all \( n \)

V. CONVERGENCE OF THE HSA ALGORITHM

In this section, we show the convergence of HSA algorithm for the case where users fix their weight coefficients \( w^n \), which is also called Rate Adaptive (RA) spectrum balancing [6] that aims at maximizing users’ rates subject to power constraint.\(^2\) Denote \( s^n_{\text{HSA}} \) as the PSD of user \( n \) in tone \( k \) at the end of iteration \( t \), where \( \sum_k s^n_{\text{HSA}} (w^n) = P^n \) is satisfied for any \( n \) and \( t \). Here one iteration is defined as one round of sequential updates of all users in Algorithm 3 (with fixed \( w^n \)). In the case of two users, we can show that

Theorem 2: The HSA algorithm globally converges to the unique fixed point in a two-user system under fixed \( w \), if

\[
\max_k \{ \alpha_k \} \max_k \{ \alpha_k \} = \frac{1}{(N-1) \sum_k \gamma (k)} < 1.
\]

The key idea behind Theorem 2 is that the HSA algorithm leads to a contraction mapping in the PSD updates, when the maximum product of the crosstalk channel gains is small enough. One extreme case is in a practical CO/RT mixed deployment case, where the crosstalk from CO to RT is negligible (i.e., \( \max_k \{ \alpha_k \} \max_k \{ \alpha_k \} = \frac{1}{(N-1) \sum_k \gamma (k)} \ll 1 \)). The value of \( \sum_k \gamma (k) \) is around 1.66 for a wide range of \( K \) (i.e., \( 32 \leq K \leq 4096 \)). The convergence result of ASB without ICI in the two-user case [5] is a special case of Theorem 2 by setting \( \gamma (k) = 0 \) for all \( k \neq 0 \). Detailed proof is omitted.

We further extend the convergence results to a system with an arbitrary \( N > 2 \) of users. In this case, we consider a more realistic but hard-to-analyze parallel updates, where time is divided into slots, and each user \( n \) updates the PSD simultaneously in each time slot according to (4) based on the PSDs in the previous slot, where the \( \lambda^n \) is adjusted such that the power constraint is satisfied. Detailed proof is omitted.

Theorem 3: Assume \( \max_m \gamma (k) \sum_{k \neq n} \alpha_k n, m \leq \frac{1}{(N-1) \sum_k \gamma (k)} \), then the HSA algorithm globally converges (to the unique fixed point) in an \( N \)-user system with fixed \( w \) and parallel updates.

Theorem 3 contains the convergence of ASB in the parallel case [5] as a special case.

VI. SIMULATION

Here we summarize a typical numerical example comparing the performances of GPS algorithm with ASB algorithm in [5]. As depicted in Fig. 2, the scenario consists of two ADSL modems, one 5 km CO deployed line, and one 3 km RT deployed line. The RT is deployed 4 km downstream from the CO. ANSI noise model A [7] has been used, which consists of 16 ISDN, 4 HDSL and 10 conventional (non-DSM capable) ADSL disturbers.

![Fig. 2. An example of mixed CO/RT deployment topology](image)

Running the ASB algorithm [5] leads to the PSD levels shown in Figure 3. Although ICI exists in the network, it is not explicitly taken into consideration in the ASB algorithm. The length of the reference line causes it to be inactive in frequencies above 0.61 MHz due to the large channel...
algorithm (from O to attained with not much complexity over standard ASB algorithm. This close to 1 Mbps performance improvement in data rate is attained with not much complexity over standard ASB algorithm. However, due to the ICI from the RT’s PSD at 0.61MHz, the CO suffers significant crosstalk and is forced to stop transmission at 0.58MHz. As a result the CO achieves 1 Mbps whilst the RT is limited to 5.3 Mbps.

Running the GPS algorithm leads to the PSD levels shown in Figure 4. Here we attempt to maximize the rate on the RT distributed line, whilst guaranteeing a target rate of at least 1 Mbps on the CO line. Comparing with figure 3, the RT has increased its PSD significantly between 0.4 MHz to 0.58 Hz, and reduced it slightly between 0.58 MHz and 0.62 MHz. The reasoning behind this choice is that whether or not the RT loads power from 0.4 MHz to 0.58 MHz, the CO will see significant crosstalk due to the ICI from the RT’s PSD at 0.61 MHz and above. For this reason the RT can increase its PSD from 0.4 to 0.58 MHz, thereby increasing its data-rate, whilst causing minimal additional degradation to the data-rate on the CO line. In a sense the ICI from the RT’s high PSD level at 0.61 MHz will mask a certain amount of crosstalk in the frequencies from 0.4 to 0.58 MHz. Hence the GPS algorithm shifts power from higher frequencies into these lower frequencies, causing minimal damage to the CO line. The small loss in rate on the CO line is easily recovered by a slight reduction in the RT PSD between 0.58 and 0.62 MHz. The result is that the CO achieves the same target rate of 1 Mbps, whilst the RT increases its achievable rate to 6.2 Mbps, a gain of 17% over ASB. This close to 1 Mbps performance improvement in data rate is attained with not much complexity over standard ASB algorithm (from \( O(KN) \) to \( O(NK^2 \log_2(K)) \)), thus push beyond the already close-to-optimal ASB performance by recognizing the inevitable asynchrony in practical DSL deployment. The comparison results are summarized in Table II.

VII. CONCLUSION

Spectrum management in asynchronous DSL networks is an under-explored research area, due to the technical difficulty introduced by inter-carrier-interferences (ICI). This paper proposes two power allocation algorithms for the asynchronous DSL network, which explicitly take asynchronous operation, and the resultant ICI, into account. The key idea behind the two algorithms is to use a reference line that is representative of a “typical” victim in the network. By setting the power spectrum level to protect the reference line, a good balance between local and global maximization can be achieved. This reference line approach exploits the specific structure of the DSL channel (e.g. the asymmetric nature of crosstalk channels in mixed CO/RT scenarios), allowing both algorithms to operate completely autonomously and with significantly lower complexity than the current state-of-the-art [1]. Convergence of the algorithms is guaranteed under reasonable conditions on the crosstalk channels which are typically satisfied in practice. The GPS version of the A-ASB algorithm achieves significant better performance than the ASB algorithm in the synchronous case.

REFERENCES


![Fig. 3. PSD allocation under ASB algorithm](image1)

![Fig. 4. PSD allocation under GPS algorithm](image2)

**TABLE II**

Performance comparison between ASB and GPS algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>CO Rate</th>
<th>RT Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASB</td>
<td>1.0 Mbps</td>
<td>5.3 Mbps</td>
</tr>
<tr>
<td>GPS</td>
<td>1.0 Mbps</td>
<td>6.2 Mbps</td>
</tr>
</tbody>
</table>