Distributed Resource Allocation Algorithms for Digital Subscriber Lines

Christopher Leung

Department of Electrical & Computer Engineering
McGill University
Montréal, Canada

May 2011

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Master of Engineering.

Abstract

The Digital Subscriber Line (DSL) environment is characterized by highly frequency-selective attenuation and by potentially large crosstalk between users. DSL resource allocation algorithms allow for an efficient use of the DSL network by managing the crosstalk-induced interference while also taking the frequency-selectivity into account. Previous resource allocation algorithms were based on worst-case situations. The more recent Dynamic Spectrum Management (DSM) resource allocation algorithms are able to constantly adapt to the channel characteristics. At one end of the DSM algorithms, there is the low-performing user-level Iterative Water-Filling (IWF) algorithm with low computational complexity and full distributivity. At the other end, the better-performing network-level algorithms are not fully distributable and have higher computational complexities.

In this thesis, an overview of the DSL environment and its resource allocation algorithms is presented. Then, three alternatives with a smaller computational load than the bisection method used in IWF's water-filling sub-algorithm are presented. These alternatives include a novel projection method for a specific case and the novel Generalized Recursive Water-Filling (GRWF) algorithm for the generalized resource allocation problem.

The Autonomous Spectrum Balancing using Multiple Reference Lines (ASB-MRL) algorithm is then presented as an algorithm capable of bridging the performance gap between the fully-distributable and low-complexity IWF, and the high-performing network-level algorithms while maintaining the benefits of each. Following that, a set of conditions on the virtual network formed by the multiple reference lines is produced to ensure that ASB-MRL allocates the resources in a near-optimal manner.

Sommaire

L'environnement des lignes d'abonné numérique (DSL) est caractérisé par un affaiblissement progressif de fréquences et une diaphonie potentiellement large entre utilisateurs. L'allocation de ressources dans le DSL permet d'utiliser le réseau DSL efficacement en gérant l'interférence produite par la diaphonie tout en prenant en compte l'affaiblissement progressif de fréquences. Les algorithmes d'allocation de ressources antérieurs étaient construits sur les principes de la pire éventualité. Plus récemment, la gestion dynamique du spectre (DSM) a permis aux algorithmes d'allocation de ressources de s'adapter continuellement aux caractéristiques des voies de transmission et cela a permis le développement de quelques algorithmes. D'un côté, il y a le remplissage d'eau itératif (IWF), un algorithme à faible complexité opérationnelle qui peut être implémenté indépendamment par chaque utilisateur. D'un autre côté, il y a les algorithmes qui gèrent tous les utilisateurs afin d'allouer les ressources beaucoup plus efficacement que le IWF. Par contre, ces algorithmes ne peuvent pas être complètement distribués parmi les utilisateurs et ils ont une plus grande complexité.

Dans ce mémoire de maîtrise, un aperçu de l'environnement DSL et des algorithmes d'allocation de ressources est introduit. Ensuite, trois alternatives pouvant s'exécuter plus rapidement que la méthode par bissection utilisée dans le sous-algorithme du IWF sont présentées. Parmi ces alternatives, une nouvelle méthode par projection est proposée pour des cas spéciaux, et une nouvelle méthode se basant sur la récursivité, le remplissage d'eau récursif généralisé (GRWF), est proposée pour les problèmes d'allocation de ressources généralisés.

L'algorithme d'équilibre de spectre autonome utilisant plusieurs lignes de référence (ASB-MRL) est ensuite présenté comme un algorithme capable d'obtenir une performance similaire aux algorithmes qui gèrent les ressources de tous les utilisateurs. Toutefois, le ASB-MRL retient les avantages du IWF: la faible complexité et l'implémentation distribué. Par la suite, un ensemble de conditions sur le réseau DSL virtuel contenant les lignes de référence est introduit pour s'assurer que le ASB-MRL alloue les ressources de façon quasi-optimale.

Acknowledgements

I would like to acknowledge my advisor Prof. Tho Le-Ngoc for taking me in as a graduate student and introducing me to the vast field of telecommunications, for his moral and financial support, and for the many interesting discussions we had. I would also like to thank Sean Huberman for all the DSL-related discussions we had and to Robert Morawski for managing and providing essential knowledge about the DSL equipments.

I also want to acknowledge the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Fonds Québecois de la Recherche sur la Nature et les Technologies (FQRNT) for their financial support.

I would also like to thank my workplace friends: Beri, Danny, Jing, Khuong, Leonardo, Loïc, Mahsa, Peng, Quang, Sanjeewa, and Soham. Thank you for the many in-and-off-topic discussions and for showing me a great time both in and out of the office.

I would also like to thank my friends Catherine, Sandrine, Emily, Zhaoyi, Frédérick, Kanitha, Puisha, for the fun times outside of school and for their constant support.

Finally, I would like to thank my parents and my family for supporting me throughout all my studies.

Contents

List of Figures

List of Tables

List of Acronyms

ANSI	American National Standards Institute
ASB	Autonomous Spectrum Balancing algorithm
ASB-DSB	ASB with MRL using DSB
ASB-MRL	General ASB with MRL
ASB-SCALE	ASB with MRL using SCALE
AWG	American Wire Gauge
CO	Central Office
dB	Decibel
dBm	Power ratio in dB using 1mW as reference
DMT	Discrete Multi Tone
DSB	Distributed Spectrum Management
DSL	Digital Subscriber Line
DSLAM	Digital Subscriber Line Access Multiplexer
DSM	Dynamic Spectrum Management
FM	Fixed Margin
GRWF	Generalized Recursive Water-Filling
ISB	Iterative Spectrum Balancing algorithm
IWF	Iterative WF algorithm
KKT	Karush-Kuhn-Tucker
MLWF	Multi-Level Water-Filling
MRL	Multiple Reference Lines
OFDM	Orthogonal Frequency-Division Multiplexing
OSB	Optimal Spectrum Balancing algorithm
PBO	Power Back-Off algorithm
PSD	Power Spectral Density
RA	Rate Adaptive
RT	Remote Terminal
SCALE	Successive Convex Approximation for Low-complExity
SIW	Selective IWF algorithm
SMC	Spectrum Management Center
SINR	Signal-to-Interference-plus-Noise Ratio

SNR	Signal-to-Noise Ratio
SSM	Static Spectrum Management
VDSL	Very high bit-rate Digital Subscriber Line
WF	Water-Filling algorithm

List of Symbols

α_k^n	Variable used by SCALE to represent $\mathrm{SINR}_k^n/(1+\mathrm{SINR}_k^n)$
a^n	Water-fill level used by user n
a_k^0	Water-fill level at which s_k transitions from inactive to active
a_k^{mask}	Water-fill level at which s_k transitions from active to saturated
a_k^{all}	Ordered combination of a_k^0 and a_k^{mask}
B	Bit-loading increment
b_k^n	Bit-loading (or spectral efficiency) on frequency tone k for user n
d_k^n	SINR with unit transmit PSD on frequency tone k for user n
Δf	Sub-carrier bandwidth
$\Delta\lambda_k^n$	Lagrange multiplier offset on frequency tone k for user n
$\Delta s_k^n(b_k^n)$	Increase in PSD required to increase b_k^n by B on frequency tone k for user n
F	Number of reference lines
$\mathcal{F}$	Set of reference lines $\{2,\ldots,F+1\}$
f_s	Discrete multi tone symbol rate
h_k^{nm}	Signal transfer function from user m to n on frequency tone k
H_k^{nm}	Power transfer function from user m to n on freqeuncy tone k
Γ	SNR gap to the Shannon capacity for all users
Γ^n	SNR gap to the Shannon capacity for user n
int_k^n	Total interference power measured by user n on frequency tone k
K	Number of tones
$\mathcal{K}$	Set of tones $\{1,\ldots,K\}$
λ^n	Lagrange multiplier for the total transmit power constraint for user n
μ_k^n	Lagrange multiplier for various constraints on frequency tone k for user n
N	Number of users
$\mathcal{N}$	Set of users $\{1,\ldots,N\}$
P_i^*	Total transmit power associated to a_i^{all}
P^n	Maximum total transmit power for user n
$P^{n,\mathrm{used}}$	Transmit power required to achieve the achievable rate
R_i^*	Achievable rate associated to a_i^{all}
R^n	Achievable rate for user n
$R^{n,\mathrm{target}}$	Target rate for user n

rec_k^n	Total received power by user n on frequency tone k
s_k^n	User n's PSD on the frequency tone k
$s_k^{n,\text{mask}}$	User n's PSD mask on the kth frequency tone
σ_k^n	Noise variance on frequency tone k seen by user n
SINR_k^n	SINR on frequency tone k for user n
υ_k^n	Lagrange multiplier for various constraints on frequency tone k for user n
W_i	Cumulative weight at iteration i used by GRWF
ω_k^n	Weight in the weighted sum bit-loading on frequency tone k for user n
ω^n	Weight in the weighted sum rate for user n
χ_k^n	Weight on the weighted sum power on frequency tone k for user n
x_k^n	Transmitted signal level on the kth frequency tone for user n
y_k^n	Received signal level on the kth frequency tone for user n
z_k^n	Additive white Gaussian noise signal on the kth frequency tone for user n

Chapter 1

Introduction

Digital Subscriber Line (DSL) is a service that allows broadband access to exist concurrently with analog voice communication over common telephone lines. DSL is attractive as it reuses the existing telephone line infrastructure thus not requiring large investments for wire installation. In fact, in 2010, DSL was the most popular broadband connection with two-thirds of the world's 485 million broadband connections being over DSL [1]. However, the telephone line infrastructures were not designed to carry high speed data transmission and therefore introduce a few factors that inhibit DSL's performance.

One inhibiting factor is the highly frequency-selective attenuation in the DSL environment. This is a property from the twisted-copper pair wires used for telephone lines and also caused by the signal reflections generated by the bridged-taps used for branching telephone lines. DSL uses Discrete Multi-Tone (DMT) transmission similar to Orthogonal Frequency-Division Multiplexing (OFDM) where the transmission frequency band is divided into many smaller sub-carriers (also called frequency tones) to approximately transform the frequency-selective broadband channel into a number of frequency-flat narrowband channels, e.g., in VDSL2, a recent DSL standard, the transmission bandwidth of 17MHz is divided into 4096 frequency tones [2].

Another factor is the overall channel attenuation of the twisted-copper pair that increases with line length. Under this factor, two telephone lines of unequal lengths will have very different attenuation levels with the longer line experiencing higher losses than on the shorter line. This results in the longer line having a much reduced achievable data rate. As a solution to this problem, Fiber-To-The-Node (FTTN) and Fiber-To-The-Curb (FTTC)

can be used where part of the twisted-copper pair is replaced with optical fiber. This effectively creates a hybrid copper-fiber network with the twisted-copper pairs covering the final few hundred meters on the client-side. Even with a hybrid copper-fiber network, the savings are still significant as the last-mile telephone infrastructures can be reused while shortening the twisted-copper line length.

The third factor is the crosstalk caused by electromagnetic coupling between different telephone lines within a same bundle. Like the channel frequency response, the crosstalk is also highly frequency selective. Moreover, the total crosstalk onto a telephone line is cumulative in that it includes the crosstalk from every other telephone line in the same bundle. This cumulative crosstalk represents a major interference source thus making DSL crosstalk-limited. Moreover, crosstalk also depends on the line lengths where the crosstalk magnitude becomes larger with line coupling length and frequency, but also attenuates over regions without coupling.

The channel response over each frequency tone can have large variations and DMT takes advantage of the variation by letting each frequency tone have its own modulation scheme to carry a different bit rate. This is where the resource allocation problem comes into play by trying to efficiently allocate the transmit power over the frequency tones. From the resource allocation problem, different objectives can be formulated such as to maximize the rate achieved over a DSL network, to minimize the power needed to achieve a set requirement [3], or to maximize the rate of certain DSL users while guaranteeing a certain quality of service on the others.

Many resource allocation algorithms were proposed to partially or fully meet at least one of the resource allocation problem objectives. At the lower performing range, there are the Static Spectrum Management (SSM) algorithms that perform the resource allocation on a single user while assuming worst-case scenario elements. However, using worst-case scenarios will produce inefficient results. Up until recently, SSM algorithms were sufficient, but with the increase in demand for higher data rates for faster file transfer rates and for applications such as VoIP and IPTV, more efficient resource allocation algorithms are required.

For a more efficient resource allocation, there are the Dynamic Spectrum Management (DSM) [4, 5, 6] algorithms that are able to constantly adapt to the channel characteristics. The DSM algorithms can be characterized as being user-level or network-level, and as being distributable or not. User-level algorithms perform resource allocation on a single-user

basis while network-level algorithms jointly optimize every user to allocate the resources as efficiently as possible. Distributive resource allocation algorithms have the benefit that they can be fully distributed to each user and implemented at the customer end. Non-distributive algorithms either have to be performed centrally at a Spectrum Management Center (SMC) or they can be semi-distributed by using a message passing system to transfer the algorithm's variables between the different users.

The most basic DSM algorithm is the fully-distributive user-level Iterative Water-Filling (IWF) algorithm [7]. Unlike the SSM algorithms which rely on worst-case scenarios, with IWF, each user generates the most efficient Power Spectral Density (PSD) given the channel gain and the interference power measured. By adapting to the channel environment, users applying IWF are able to produce achievable data transmission rates up to 100% over SSM algorithms [8, 9]. Yet, even if IWF does produce a user-efficient resource allocation, it does not produce a network-efficient resource allocation. For optimal results, network-level algorithms that are able to jointly manage every user are required. However, network-level algorithms have a computational complexity higher than that of IWF. And, as previously mentionned, network-level algorithms must also be either performed centrally or use a message passing system between users to exchange the algorithm's variables.

1.1 Thesis Contributions

In the first part, this thesis proposes an algorithm to reduce the computational load of the water-filling sub-algorithm in IWF. A projection method that can perform water-filling in a single iteration in the best case scenario is proposed for the special case where no PSD mask is used. Then, a generalization on the resource allocation problem for both rate maximization and for power miniminization is introduced. Using the generalized problem, the corresponding generalized water-filling solution is derived, and a recursive approach for finding the total transmit power used and the achievable rate is proposed to finally produce the Generalized Recursive Water-Filling (GRWF) algorithm. The GRWF algorithm is capable of performing water-filling within a discrete number of iterations determined only by the number of frequency tones while having a smaller computational footprint than other methods. Finally, a discrete bit-loading approach is proposed for the water-filling-like network-level algorithms.

In the second part, this thesis looks at modifying IWF in order to make it perform

close to network-level algorithm thus bridging the gap between user-level and network-level resource allocation algorithms. Using effective water-filling levels, an analysis between IWF and network-level algorithms is performed and a modification to IWF under the form of a Lagrange multiplier offset is proposed. With the Lagrange offset, the modified IWF algorithm has the potential to produce network-like results. The concept of using multiple reference lines is then introduced as a method for producing the offsets. This results in the introduction of the Autonomous Spectrum Balancing using Multiple Reference Lines (ASB-MRL) algorithms as user-level algorithms that optimize over a virtual network. Through the virtual network, the ASB-MRL algorithms are capable of achieving similar performances as network-level algorithms while requiring fewer global iterations to converge and having the same computational complexity as IWF. Then, three conditions on the virtual network containing the multiple reference lines are developed such that the ASB-MRL algorithms can consistently produce network-level-like results. Finally, a Monte Carlo style simulation is performed over two typical types of DSL networks to show the network-level-like performance capabilities of the ASB-MRL algorithms.

1.2 Thesis Outline

The remainder of this thesis is organized as follows. Chapter 2 presents the background necessary for understanding the topics discussed in this thesis. This includes introducing the DSL environment and the crosstalk-limited DSL network. Then the system model used throughout this thesis is presented and the DSL resource allocation problem is formulated. Finally, this chapter describes previous user-level and network-level DSL resource algorithms and discusses their main differences.

Chapter 3 begins by reviewing the bit-loading approach and proposes the projection method as an alternative to bisection for finding the water-filling level in the water-filling algorithm in order to reduce the computational load of the IWF algorithm. After showing that both approaches are only suitable for specific cases, the chapter introduces the GRWF algorithm by first generalizing the resource allocation used by water-filling, then by finding a recursive formulation for determining the total transmit power required and the achievable rate obtained at different discrete water-filling levels, and finally by formulating a way for using the recursive formulation over a continuous set of water-filling levels. The GRWF algorithm is then compared to the other water-filling algorithms by determining

their computational loads. Finally, a bit-loading method for water-filling-like algorithms is introduced.

Chapter 4 begins by differentiating the sub-optimal IWF and optimal network-level DSL resource allocation algorithms by comparing the effective water-filling levels they produce. Using the differences, an extension to the IWF is introduced through the form of a Lagrange multiplier offset that gives it the potential to perform near-optimally. Then, the concept of using multiple reference lines is introduced as a method for generating the Lagrange multiplier offsets and two ASB-MRL algorithms are derived: ASB-DSB and ASB-SCALE. A relaxation is then performed on the ASB-MRL algorithms to reduce their computational complexity per iteration to that of IWF. After, three conditions on the set of reference lines used are formed in order to have ASB-MRL consistently producing near-optimal results. Finally, an illustrative example is given through a Monte Carlo style simulation over two different types of DSL networks.

Chapter 5 summarizes this thesis and proposes future research direction from the work accomplished.

Chapter 2

DSL Resource Allocation Algorithms[1]

The Digital Subscriber Line (DSL) system allows for the reuse of plain old telephone system (POTS) infrastructures for data transmission through twisted copper pairs. However, POTS were initially designed for voice communication in the 0 to 4 kHz range and not for the MHz range used by DSL thus creating many challenges to DSL in the form of frequency-selective channels and potentially large crosstalk interference. In order to address these challenges, various algorithms and techniques have been introduced.

This chapter introduces the DSL channel characteristics and resource allocation algorithms. In Section 2.1, a brief overview of the DSL environment is presented. Then, Section 2.2 talks about the crosstalk-limited DSL network. The system model used throughout this thesis is given in Section 2.3 and the DSL problems in Section 2.4. Finally, various DSL resource allocation algorithms are described in Section 2.5.

2.1 Discrete Multi-Tone

The DSL system allows data communication to work alongside with voice communication by using the higher frequency bands (>25kHz) in the POTS while voice communication uses the lower frequency band (<25kHz). However, the POTS infrastructure was not initially designed for data communication over the higher frequency bands where the channel is highly frequency-selective. This frequency selectiveness is characterized by an attenuation that increases with frequency. Additionally, the attenuation also increases with the twisted

[1]Part of Chapter 2 has been presented in [10].

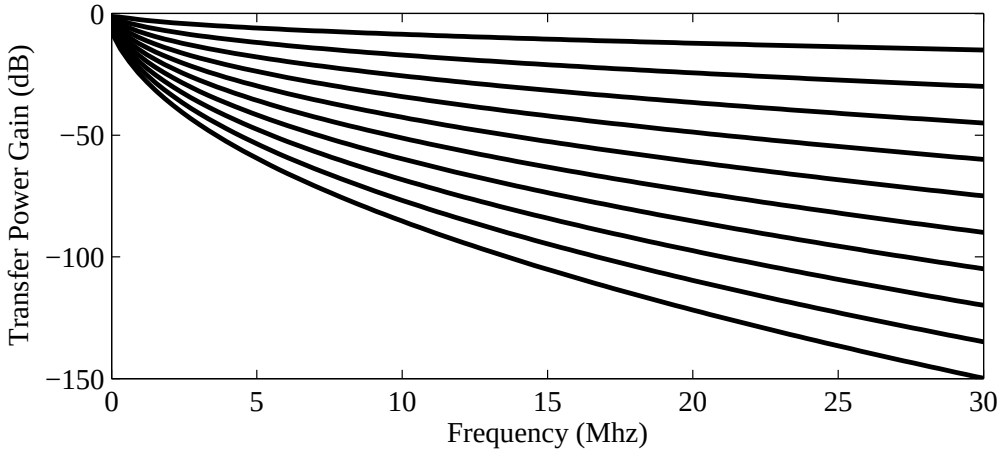


Fig. 2.1 Channel gain for various line lengths. The highest channel gain corresponds to a line length of 100m. From the highest channel gain towards the lowest channel gain, the line length increases by 100m until it reaches 1000m.

copper pair line length. Figure 2.1 plots a few channel gains over the frequency band used by VDSL2, a recent DSL standard, for lengths varying from 100m to 1000m. The channel gains were generated using the channel model specified in [11] which corresponds to an ideal wire and do not include additional attenuations that can be caused by other factors including wire splitting.

In order to cope with the frequency-selectiveness of the twisted copper pairs, VDSL2 uses Discrete Multi-Tone (DMT) as the transmission technique. This technique is similar to OFDM where the bandwidth is divided into many orthogonal sub-carriers, called frequency tones in DMT. By dividing the whole frequency-selective bandwidth, each frequency tone can be made to seem less frequency-selective and even non frequency-selective with a sufficient number of divisions. The division results in many frequency tones each with their own channel gain as shown in Figure 2.2. Obviously some frequency tones will have better channel gains than others and DMT can benefit from the variability by allowing each frequency tone to have a different modulation scheme customized for each tone's potential spectral efficiency . Thus, with varying modulation schemes, DMT can transmit more bits over the better frequency tones and ignore the ones with very poor channel gains. On the other hand, even with the severe frequency-selective channels, the DSL environment has the benefit of being very slow fading.

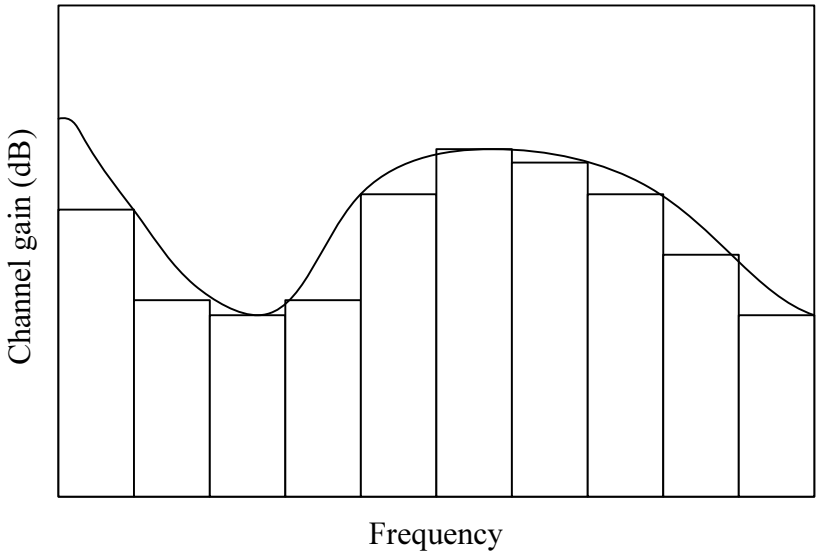


Fig. 2.2 From frequency-selective to many non frequency-selective sub-carriers.

2.2 Crosstalk

Another challenge that DSL is faced with is the presence of crosstalk. Because the twisted copper pairs in POTS are bundled together, each pair can introduce high levels of electromagnetic interference onto another pair. Figure 2.3 shows the two kinds of crosstalk that can exist: far-end crosstalk (FEXT) from one end of one pair to the other end of the other pair, and near-end crosstalk (NEXT) between the same ends of two pairs. Of the two crosstalk sources, only FEXT remains in VDSL2 as NEXT is prevented by using frequency division duplex in which the frequency tones are divided into either upstream of downstream transmission on every copper pair. The transmission direction used for VDSL2 co-existing with voice communication in North America is summarized in the bandplan given in Figure 2.4 [11]. The upstream direction corresponds to the transmission from the customer's modem towards the Digital Subscriber Line Access Multiplexer (DSLAM) or the telephone Central Office (CO). The downstream direction corresponds to the opposite, from the CO to the customer. Note that the bandplan allocates a much larger frequency range to downstream as the standard gives much more priority towards data downloading than uploading.

On a pair-to-pair basis, FEXT may not represent a large issue. However, twisted copper pair bundles may contain up to 25 pairs and bundles can be combined to create even larger

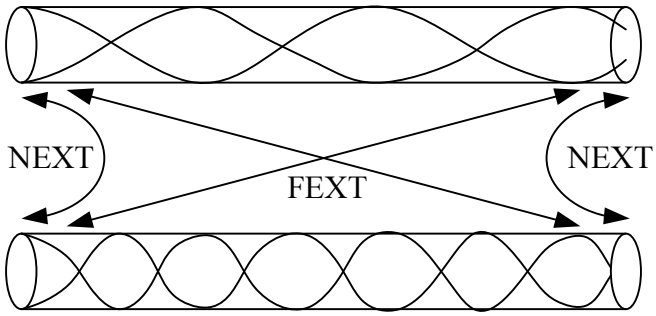


Fig. 2.3 Crosstalk between two twisted copper pairs.

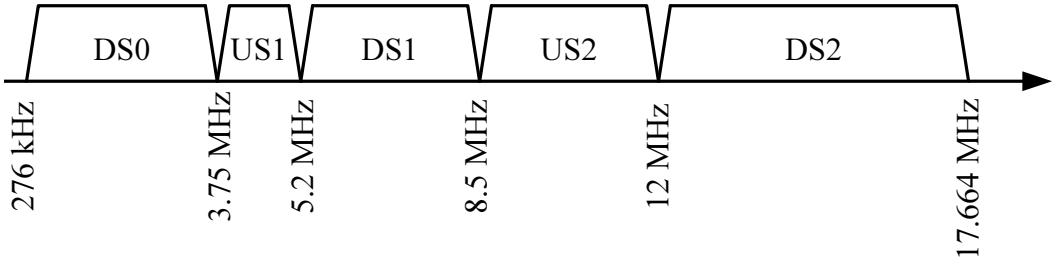


Fig. 2.4 VDSL bandplan with DS denoting downstream and US for upstream.

bundles containing up to 200 pairs. Each additional pair increases the total crosstalk on a given pair and can represent a significant source of interference resulting in crosstalk-limited performances over DSL networks.

2.3 System Model

With DMT using orthogonal sub-carriers with approximately no frequency-selectiveness, we assume that each frequency is free of inter-carrier interference and inter-symbol interference. Then, the DSL signal model over each frequency tone is presented in Figure 2.5 where N represents the number of twisted copper pairs, or users, x^n the nth transmitter, y^n the nth receiver, and z^n the additive noise at the nth receiver which includes thermal noise, alien crosstalk, and radio frequency interference. For the general kth frequency tone, the model is formulated as

$$y_k^n = z_k^n + \sum_{m \in \mathcal{N}} h_k^{nm} x_k^m$$

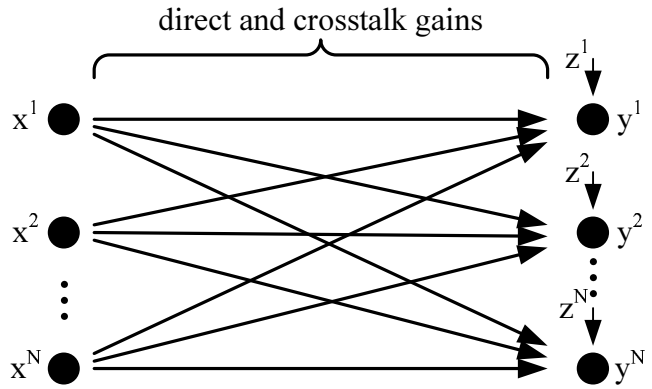


Fig. 2.5 DSL signal model.

where h_k^{nm} represents the transfer signal gain from transmitter m to receiver n over the kth frequency tone. The set $\mathcal{N}$ represents the set of users and contains $\{1, 2, \ldots, N\}$. Similarly, the set $\mathcal{K}$ represents the set of frequency tones and contains $\{1, 2, \ldots, K\}$ where K is the number of frequency tones.

The transmit Power Spectral Density (PSD) for user n on frequency tone k is denoted as s_k^n and is given by $\mathrm{E}\{|x_k^n|^2\}/\Delta f$ where $\mathrm{E}\{\cdot\}$ represents the expectation function and Δf the frequency tone spacing. Similarly, the transfer power gain is defined as $H_k^{nm} \triangleq |h_k^{nm}|^2$ and the noise power spectral density as $\sigma_k^n \triangleq \mathrm{E}\{|z_k^n|^2\}/\Delta f$.

The total interference power received by user n on frequency tone k is defined as

$$\mathrm{int}_k^n \triangleq \sigma_k^n + \sum_{\substack{m \in \mathcal{N} \\ m \neq n}} H_k^{nm} s_k^m$$

and the total received power as

$$\mathrm{rec}_k^n \triangleq \sigma_k^n + \sum_{m \in \mathcal{N}} H_k^{nm} s_k^m = \mathrm{int}_k^n + H_k^{nn} s_k^m.$$

With a sufficient number of users and by assuming that there is no useful information in the received crosstalk, the interference can be modeled as a Gaussian distribution and the spectral efficiency, or bit-loading, is

$$b_k^n = \log_2 \left(1 + \frac{1}{\Gamma} \frac{H_k^{nn} s_k^n}{\mathrm{int}_k^n} \right) \tag{2.1}$$

where Γ represents the Signal to Noise Ratio (SNR) gap and is calculated using the desired Bit Error Rate (BER), noise margin, and coding gain. The total sum rate for user n is obtained by using

$$R^n = f_s \sum_{k \in \mathcal{K}} b_k^n \tag{2.2}$$

where f_s represents the DMT symbol rate.

2.3.1 Channel Model

The channel model used throughout this thesis is based on the ANSI standard [11]. In this model, the direct transfer power gains are based on the RLCG transmission model. On the other hand, the crosstalk power gain model is based on empirical measurements and is made such that it represents a 99% worst-case scenario. That is, there is a probability of 99% that the actual crosstalk power gain is inferior to the one calculated by the model.

The additive noise z_k^n is assumed to be from a zero-mean Gaussian distribution with variance σ_k^n and in the simulations it is set to a constant -140dBm following the ANSI standard [11].

2.4 DSL Resource Allocation

The purpose of DSL resource allocation is to make a DSL network as efficient as possible whether by maximizing the network sum rate or by minimizing the network sum power. Over time, many algorithms were developed ranging from the early single-user static algorithms to the more recent multi-user dynamic algorithms. The static and dynamic terms refer to the algorithm's ability to adapt in a changing network. This section lays out the objective functions used by the algorithms presented in this thesis and by most other DSL resource allocation algorithms.

2.4.1 Objective Function

This thesis covers two main types of DSL resource allocation:

1. Rate Adaptive (RA) optimization where the purpose is to maximize the overall sum rate of the network given a certain power constraint, and

2. Fixed Margin (FM) optimization where the purpose is to minimize the overall sum power of the network given a certain minimum rate constraint.

Both RA and FM optimization can be combined into a mixed RA-FM case where the goal is to maximize the sum rate of a given group of users while still guaranteeing a minimum rate on the other users. With some algorithms, the solution to the RA case is very similar to the FM case thus allowing for a mixed RA-FM algorithm to be easily created.

Rate Adaptive Optimization

The optimization problem used by RA optimization is as follows:

$$\begin{aligned} \max_{s_k^n \forall k \in \mathcal{K}, n \in \mathcal{N}} \quad & \sum_{n \in \mathcal{N}} \omega^n R^n \\ \text{subject to} \quad & \Delta f \sum_{k \in \mathcal{K}} s_k^n \leq P^n \quad , \forall n \in \mathcal{N} \\ & 0 \leq s_k^n \leq s_k^{\text{mask},n} \quad , \forall k \in \mathcal{K}, n \in \mathcal{N} \end{aligned} \tag{2.3}$$

where P^n refers to the maximum allowable total transmit power used by user n, ω^n is the weight of user n representing its maximization priority, and $s_k^{\text{mask},n}$ is the maximum PSD that user n can use on frequency tone k.

Fixed Margin Optimization

The FM case consists of minimizing the network sum power and it is summarized mathematically as follows:

$$\begin{aligned} \max_{s_k^n \forall k \in \mathcal{K}, n \in \mathcal{N}} \quad & \sum_{\substack{k \in \mathcal{K} \\ n \in \mathcal{N}}} -s_k^n \\ \text{subject to} \quad & R^n \geq R^{n,\text{target}} \quad , \forall n \in \mathcal{N} \\ & 0 \leq s_k^n \leq s_k^{\text{mask},n} \quad , \forall k \in \mathcal{K}, n \in \mathcal{N} \end{aligned} \tag{2.4}$$

where $R^{n,\text{target}}$ corresponds to the the minimum rate that user n must achieve.

Mixed RA-FM Optimization

In the mixed RA-FM case, each user must be categorized as either being an RA user or an FM user. The mixed RA-FM optimization will then maximize the sum rate of the RA users while ensuring that the FM users meet their respective minimum target rate. Letting $\mathcal{RA}$ denote the set containing the RA users and $\mathcal{FM}$ for the FM users, then the mixed RA-FM optimization problem is summarized as follows:

$$\begin{aligned} \max_{s_k^n \forall k \in \mathcal{K}, n \in \mathcal{N}} \quad & \sum_{n \in \mathcal{RA}} \omega^n R^n \\ \text{subject to} \quad & \Delta f \sum_{k \in \mathcal{K}} s_k^n \leq P^n \quad , \forall n \in \mathcal{N} \\ & R^n \geq R^{n,\text{target}} \quad , \forall n \in \mathcal{FM} \\ & 0 \leq s_k^n \leq s_k^{\text{mask},n} \quad , \forall k \in \mathcal{K}, n \in \mathcal{N}. \end{aligned} \tag{2.5}$$

Note that the RA problem is a special case of the mixed RA-FM problem where all users are members of the set $\mathcal{RA}$ and the set $\mathcal{FM}$ is empty.

2.4.2 Other Objective Functions

Aside from RA and FM, another main type of objective function exists and it consists of maximizing the SNR gap in order to improve the stability of the user's line to other sources of interference and has been used by [12, 13, 14] . This type of optimization is also called margin optimization and often has the following form:

$$\begin{aligned} \max_{s_k^n \forall k \in \mathcal{K}, n \in \mathcal{N}} \quad & \sum_{n \in \mathcal{N}} \omega^n \Gamma^n \\ \text{subject to} \quad & \Delta f \sum_{k \in \mathcal{K}} s_k^n \leq P^n \quad , \forall n \in \mathcal{N} \\ & R^{n,\text{margin}} \geq R^{n,\text{target}} \quad , \forall n \in \mathcal{N} \\ & 0 \leq s_k^n \leq s_k^{\text{mask},n} \quad , \forall k \in \mathcal{K}, n \in \mathcal{N} \end{aligned}$$

with the user's sum rate now defined as

$$R^{n,\text{margin}} = \Delta f \sum_{k \in \mathcal{K}} \log_2 \left(1 + \frac{1}{\Gamma^n} \frac{H_k^{nn} s_k^n}{\text{int}_k^n} \right).$$

Other objective functions can also exist to achieve different goals by usually using a different utility in the objective function.

2.5 DSL Resource Allocation Algorithms

There are many algorithms that attempt to solve the DSL resource allocation problems to various degrees. At the lower end, there are the Static Spectrum Management (SSM) algorithms that use very simple methods that result in far-from-optimal solutions. Then, there are the Dynamic Spectrum Management (DSM) algorithms that can continuously adapt to a changing channel and that can produce user-level optimal results and network-level (or multi-user) optimal results. This section presents a few SSM and DSM algorithms and goes more in depth into a few DSM algorithms that will be used throughout this thesis.

2.5.1 Static Spectrum Management

The SSM algorithms, also referred to as DSM level-0 [15], are characterized by their inability to adapt to a given DSL network. Instead, SSM algorithms are designed to cover most cases by using a worst-case scenario system.

Noise Reference

The noise reference method sets a user's PSD such that crosstalk it produces onto a reference user is equal to some noise reference. The equation equivalent is

$$s_k^n = \frac{\text{ref\_noise}}{H_k^{nm}}$$

for setting the PSD on frequency tone k for user n with m acting as the reference user. This algorithm tries to limit the total interference by limiting the crosstalk contribution from every user. Assuming a network with N identical users, and reference user m being modeled after user n, then the bit-loading will effectively become

$$\begin{aligned} b_k^n &= \log_2\left(1 + \frac{1}{\Gamma}\frac{H_k^{nn}\text{ref\_noise}/H_k^{nm}}{\sigma_k^n + (N-1)\text{ref\_noise}}\right) \\ &\approx \log_2\left(1 + \frac{1}{\Gamma}\frac{H_k^{nn}}{(N-1)H_k^{nm}}\right) \end{aligned}$$

where the crosstalk is assumed to be much greater than the noise variance. From the above approximation, it is clear that the performance from Noise Reference tightly depends on H_k^{nm}. However, H_k^{nm} is a worst-case crosstalk gain that restricts the algorithm from achieving better performances in non-worst-case scenarios.

Flat PBO

Flat Power Back-Off (PBO) consists of generating a flat PSD for each user and of adjusting the PSD level until a desired target rate is obtained, a similar process as in the FM problem. Thus, Flat PBO solves

$$\begin{aligned} & R^n = R^{n,\text{target}}, \quad \forall n \in \mathcal{N} \\ \text{where} \quad & \\ & s_1^n = s_2^n = \ldots = s_K^n, \quad \forall n \in \mathcal{N}. \end{aligned} \tag{2.6}$$

Although quite simplistic, Flat PBO is able to solve a challenge in DSL networks: the near-far problem. In its simplest case, the near-far problem arises when users of different line lengths are all connected to the same CO. In either transmission direction, the direct channel gains are more attenuated on the furthest user and less on the nearer user as previously shown in Figure 2.1. More importantly, in the upstream direction, from the customer's modem towards the CO, the crosstalk channel gain from the near user's transmitter to the far user's receiver is much greater than that from the far user's transmitter to the near user's receiver. This is because the far user's PSD has had time to attenuate before reaching the crosstalk zone where both copper pairs are bundled together. Thus, if both near and far users were allowed to transmit over the same frequency tones using the same PSD level, then the near user will generate too much crosstalk towards the far user. This is depicted on the upper half of Figure 2.6 where the width of the arrows illustrates the power attenuation over distance. Clearly, the near user with its better direct channel gain does not need to transmit using a large PSD in order to obtain the same bit-loading as the far user. Thus, with Flat PBO, the near user reduces its PSD such that both near and far users can obtain a similar SNR. This reduction by the near user is depicted in the lower half of Figure 2.6 where the width of the arrows show that the received crosstalk on both users is now more balanced.

On the downstream direction, Flat PBO is not usually needed if all users are connected

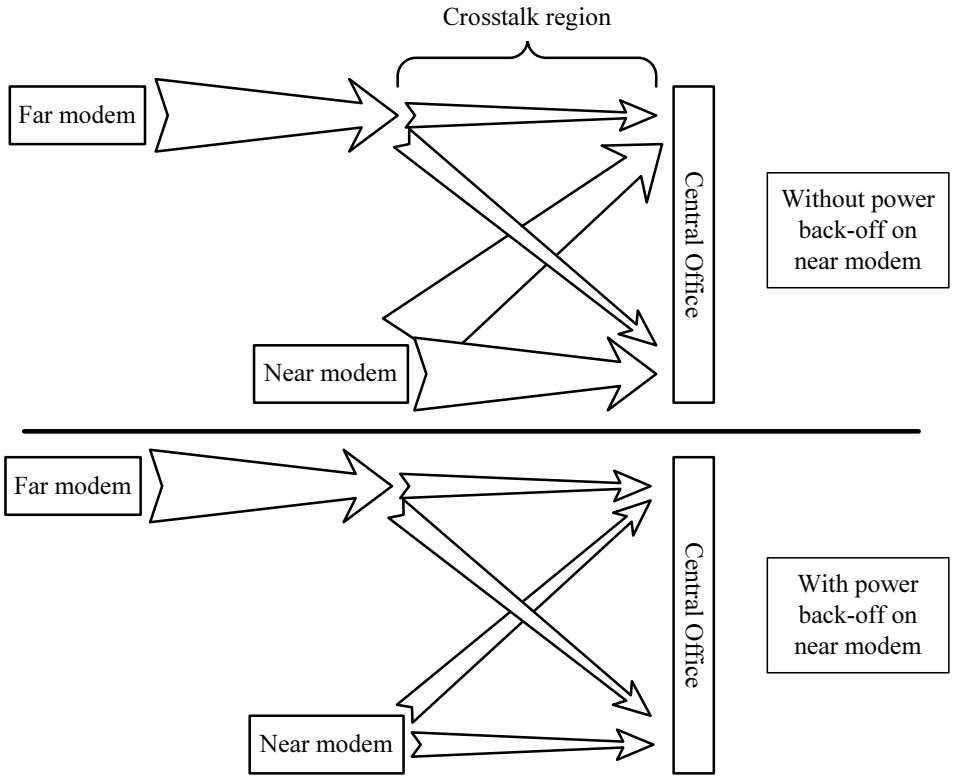


Fig. 2.6 Near-far effect and PBO example showing the effects of not reducing a near-user's upstream total transmit power. The width of the arrows represent the signal power level.

to the same CO. This is due to both crosstalk and direct channel gains attenuating at the same time.

Although simple to calculate, Flat PBO's main disadvantage is its flat PSD does not optimally use the available spectrum. Since each frequency tone has a different channel response and offers different potential spectral efficiency, it is better to use a varying PSD that puts more power into the frequency tones with a better channel gain and lower interference and less in the worser frequency tones.

2.5.2 User-Level DSM Algorithms

Unlike SSM, user-level DSM algorithms are able to dynamically adapt the PSD to the varying channel gains over different frequency tones. Unlike their more complex counterparts, the multi-user or network-level DSM algorithms, user-level algorithms do not require any

coordination with other users for the PSD generation. Thus, user-level algorithms usually have the potential to be fully-distributed.

Iterative Water-Filling

The Iterative Water-Filling (IWF) for DSL [7] is a rather simplistic algorithm where each user applies the optimal PSD given the interference it measures. This optimal PSD is obtained using a sub-algorithm called water-filling derived in [16] and each user continuously performs it until their PSDs converge to an equilibrium.

The water-filling rate maximization procedure works by having each user n solve the following optimization problem:

$$\begin{aligned} \max_{s_k^n, k\in\mathcal{K}} \quad & R^n \\ \text{such that} \quad & \Delta f \sum_{k\in\mathcal{K}} s_k^n \leq P^n \\ & 0 \leq s_k^n \leq s_k^{\text{mask},n}. \end{aligned} \tag{2.7}$$

The water-filling problem has a concave objective function and its feasibility region is in a convex set, thus meeting the Karush-Kuhn-Tucker (KKT) conditions is sufficient to obtain a globally-optimal solution [17]. To meet the KKT conditions, Lagrange multipliers are used in order to incorporate the constraints into the objective function. This gives the following Lagrangian user n:

$$\mathcal{L}^n = R^n - \lambda^n(\sum_{k\in\mathcal{K}} s_k^n - P^n/\Delta f) + \sum_{k\in\mathcal{K}}(\upsilon_k^n s_k^n - \mu_k^n(s_k^n - s_k^{\text{mask},n})).$$

The Lagrangians λ^n, υ_k^n, and μ_k^n are null when their respective constraints are 0 and positive when not met. Solving its derivative and isolating s_k^n gives

$$s_k^n = \frac{1/\ln(2)}{\lambda^n + \mu_k^n - \upsilon_k^n} - \frac{\Gamma\text{int}_k^n}{H_k^{nn}}$$

which is equivalent to

$$s_k^n = \left[a^n - \frac{\Gamma\text{int}_k^n}{H_k^{nn}}\right]_0^{s_k^{\text{mask},n}} \tag{2.8}$$

where a^n is a non-negative number that represents the water-filling level and $[\cdot]_a^b$ bounds the argument between a and b. The water-filling level is set such that the total transmit power constraint $\Delta f \sum_{k\in\mathcal{K}} s_k^n \leq P^n$ becomes tight.

The same water-filling solution is found when minimizing the power in the following optimization problem

$$\begin{aligned} \max_{s_k^n, k\in\mathcal{K}} \quad & -\sum_{k\in\mathcal{K}} s_k^n \\ \text{such that} \quad & R^n \geq R^{n,\text{target}} \\ & 0 \leq s_k^n \leq s_k^{\text{mask},n} \end{aligned} \tag{2.9}$$

by again solving for the KKT conditions.

In the IWF algorithm, each user iteratively applies (2.8) until s_k^n converges. IWF can perform well on some scenarios and particularly well in DSL networks where all bundled users are connected to the CO in downstream transmission. In upstream transmission, IWF falls into the same problem solved by Flat PBO where near users must reduce the total transmit power they used.

The main computation challenge faced by IWF lies in the water-filling procedure in which the water-filling level needs to be found. One widely used method to find the level is by using a bisection approach over a set initially bounded below by 0 and above by a sufficient large number to cover all cases. At every bisection step, the mean of the boundary is used as the level and the total transmit power used is calculated. If the total transmit power is below the maximum, then the tested level is chosen as the new lower bound. Similarly, if the total transmit power is above the maximum, then the tested level becomes the new upper bound. However, bisection is costly in that it can require many steps to reach a given error margin. Other methods for finding the water-filling level are introduced in Chapter 3.

Autonomous Spectrum Balancing

The Autonomous Spectrum Balancing (ASB) algorithms developed in [18, 19, 20] is given the potential to perform better than IWF by adding the notion of a reference line. The

addition of a reference line transforms the user-centric optimization problem for user n into

$$\begin{aligned} \max_{s_k^n, k\in\mathcal{K}} \quad & f_s \sum_{k\in\mathcal{K}} \left(\omega^n \log_2 \left(1 + \frac{1}{\Gamma} \frac{H_k^{nn} s_k^n}{\text{int}_k^n} \right) + \omega^{\text{ref}} \log_2 \left(1 + \frac{1}{\Gamma} \frac{H_k^{\text{ref,ref}} s_k^{\text{ref}}}{\sigma_k^{\text{ref}} + H_k^{\text{ref},n} s_k^n} \right) \right) \\ \text{such that} \quad & R^n \geq R^{n,\text{target}} \\ & 0 \leq s_k^n \leq s_k^{\text{mask},n} \end{aligned} \tag{2.10}$$

where ref denotes the reference line.

In [18], the solution required solving a cubic equation at every iteration. In order to remove this cubic equation need, [20] proposed a method that simply required the user to iteratively update its PSD using the following:

$$s_k^n = \left[\frac{1}{\lambda^n + \omega^{\text{ref}} H_k^{\text{ref},n} \left(\frac{1}{\sigma_k^{\text{ref}} + H_k^{\text{ref},n} s_k^n} - \frac{1}{\sigma_k^{\text{ref}} + H_k^{\text{ref},n} s_k^n + H_k^{\text{ref,ref}} s_k^{\text{ref}}} \right)} - \frac{\Gamma \text{int}_k^n}{H_k^{nn}} \right]_0^{s_k^{\text{mask},n}} \tag{2.11}$$

where the normalized Lagrange multiplier λ^n is found at every iteration such that the total transmit power constraint $R^n \geq R^{n,\text{target}}$ tight. If the resulting λ^n is negative, then it is set to 0. The update formula in (2.11) is actually a special case of the Autonomous Spectral Balancing using Multiple Reference Lines developed in Chapter 4 and a more precise derivation is provided there.

Most of the challenges faced by ASB lies in the choice of the reference line parameters $H_k^{\text{ref,ref}}$ and s_k^{ref} over all frequency tones. One potential method is to make the reference line represent the worse user in a DSL network which is often the user with the longest line lengths. Under this interpretation, ASB works by protecting the reference line and it is this method that is tested in [18]. However, often times, choosing the longest line does not always give good performance and can even make ASB perform worse than IWF. In order to remedy this, [21] proposed a method where a Spectrum Management Center (SMC) generates the reference line information at every iteration by using the actual crosstalk values between the users. However, using this method requires a message passing system between the users and the SMC similar to that use by network-level algorithms thus removing the ability for the algorithm to be fully distributive.

2.5.3 Network-Level Optimization

The network-level optimization algorithms are the ones that actually attempt to solve the full RA (2.3), FM (2.4), and (2.5) problems. The network-level algorithms can be further divided into more groups: globally-optimal algorithms, locally-optimal algorithms, and near-optimal algorithms. The difference in all types lies in whether the algorithm can produce a result that is globally optimal over the entire feasible region, that is locally optimal in a subset of the feasible region, or if the solution is non-optimal, yet giving results in the same range as the optimal algorithms.

Optimal Spectrum Balancing

The Optimal Spectrum Balancing (OSB) [22] is a globally-optimal algorithm that creates a dual problem to the RA problem in order to perform an exhaustive search over a smaller span. It begins by introducing a Lagrange multiplier to the RA problem and produces the following Lagrangian:

$$\mathcal{L} = \sum_{n\in\mathcal{N}} \omega^n R^N - \sum_{n\in\mathcal{N}} \lambda^n \Big(\sum_{k\in\mathcal{K}} s_k^n - P^n/\Delta f \Big) \tag{2.12}$$

where the Lagrange multiplier λ^n is a non-negative number. Normally, the exhaustive search will take place in two dimensions and must be performed for every set of λ^n tried. Hence, for each λ^n tried, q^{NK} variables where q is the number of discrete values used to represent s_k^n. Yet, instead of searching through q^{NK} variables, the OSB method then decomposes the Lagrangian into a sum over the frequency tones

$$\mathcal{L} = \sum_{k\in\mathcal{K}} \Big(\sum_{n\in\mathcal{N}} \omega^n f_s b_k^n - \sum_{n\in\mathcal{N}} \lambda^n s_k^n \Big).$$

This effectively reduces the search space by separating the problem into K subproblems that only need to search through every combination of s_k^n for all n. Hence, the dual problem reduces the exhaustive search complexity from $\mathcal{O}(e^{KN})$ to $\mathcal{O}(Ke^N)$. The global-optimality of the result was proven in [23] for the case where the number of frequency tones goes to infinity. An FM version was presented in [24].

Iterative Spectrum Balancing

The Iterative Spectrum Balancing (ISB) algorithm was first presented in [25] and further developed in [26]. ISB is a near-optimal algorithm that further reduces the complexity in OSB by performing the exhaustive search over each user one at the time. That is, instead of maximizing over every users at the same time, it forces all the users but one to make their PSD constant then it performs exhaustive search on the sole user with a non-constant PSD. This has the ability to reduce the computational runtime in the order of days with OSB to within the order of minutes. However, since the exhaustive search is now performed on one dimension at a time, the global optimality can no longer be guaranteed hence the near-optimal classification.

Other Globally-Optimal Algorithms

OSB and ISB were among the first algorithms that are able to produce results within the optimal range. Since then, various other optimization methods have been applied towards solving the RA problem from using geometric programming in [27] to using monotonic optimization in [28, 29] or using difference of convex programming in [30]. All three methods for optimization can generate globally-optimal solutions, but like OSB, have a long computational time. Moreover, techniques for reducing the number of iterations required for convergence in OSB and ISB have also been developed in [31, 32] but still do not make either algorithm tractable for DSL networks with many users. On the other hand, the next two algorithms decrease the computational complexity but at the tradeoff of producing locally-optimal results.

SCALE

The Successive Convex-Approximation Low complExity (SCALE) [33] is a locally-optimal algorithm that works by approximating the bit-loading using a lower-bounding concave function. This effectively transforms the RA problem from a harder difference of convex problem into an easier concave maximization problem. It was shown in [33] that the lower-bounding function used is

$$\alpha \log_2 z + \beta \leq \log_2(1+z)$$

and that it becomes tight at z_0 when choosing α and β to be

$$\alpha = \frac{z_0}{1+z_0}$$
$$\beta = \log_2(1+z_0) - \frac{z_0}{1+z_0}\log_2 z_0.$$

In order to make this lower bound on the bit-loading concave, a change in variable space from s_k^n to $\tilde{s}_k^n$ where $s_k^n = e^{\tilde{s}_k^n}$ is required. This gives the following bit-loading concave lower bound:

$$\tilde{b}_k^n = \alpha_k^n \left(\log_2 \left(\frac{H_k^{nn}}{\Gamma} \right) + \frac{\tilde{s}_k^n}{\ln 2} - \log_2 \left(\sigma_k^n + \sum_{\substack{m \in \mathcal{N} \\ m \neq n}} H_k^{nm} e^{\tilde{s}_k^m} \right) \right) + \beta_k^n \leq b_k^n$$

where the log-sum-exponential is a concave function. Using this concave lower bound, the Lagrangian of mixed RA-FM problem is

$$\mathcal{L} = \sum_{n \in \mathcal{RA}} \omega^n \tilde{R}^n - \sum_{n \in \mathcal{N}} \lambda^n \left(\sum_{k \in \mathcal{K}} s_k^n - \frac{P^n}{\Delta f} \right) - \sum_{n \in \mathcal{FM}} \mu^n \left(R^{n,\text{target}} - \tilde{R}^n \right)$$

where $\tilde{R}^n$ is the concave lower bound to the rate and is given by

$$\tilde{R}^n = f_s \sum_{k \in \mathcal{K}} \tilde{b}_k^n.$$

Normalizing the Lagrangian to remove the $\ln 2$ factor then taking its derivative over $\tilde{s}_k^n$ gives

$$\frac{\partial \mathcal{L}}{\partial \tilde{s}_k^n} = \omega^n f_s \alpha_k^n - \lambda^n e^{\tilde{s}_k^n} - \sum_{\substack{m \in \mathcal{RA} \\ m \neq n}} \frac{\omega^m f_s \alpha_k^m H_k^{mn} e^{\tilde{s}_k^n}}{\sum_{\substack{m \in \mathcal{N} \\ j \neq m}} H_k^{mj} e^{\tilde{s}_k^j}} - \sum_{\substack{m \in \mathcal{RA} \\ m \neq n}} \frac{\mu^m f_s \alpha_k^m H_k^{mn} e^{\tilde{s}_k^n}}{\sum_{\substack{m \in \mathcal{N} \\ j \neq m}} H_k^{mj} e^{\tilde{s}_k^j}} \qquad (\text{if } n \in \mathcal{RA})$$

$$\frac{\partial \mathcal{L}}{\partial \tilde{s}_k^n} = \mu^n f_s \alpha_k^n - \lambda^n e^{\tilde{s}_k^n} - \sum_{\substack{m \in \mathcal{RA} \\ m \neq n}} \frac{\omega^m f_s \alpha_k^m H_k^{mn} e^{\tilde{s}_k^n}}{\sum_{\substack{m \in \mathcal{N} \\ j \neq m}} H_k^{mj} e^{\tilde{s}_k^j}} - \sum_{\substack{m \in \mathcal{RA} \\ m \neq n}} \frac{\mu^m f_s \alpha_k^m H_k^{mn} e^{\tilde{s}_k^n}}{\sum_{\substack{m \in \mathcal{N} \\ j \neq m}} H_k^{mj} e^{\tilde{s}_k^j}} \qquad (\text{if } n \in \mathcal{FM}). \tag{2.13}$$

Notice that the Lagrange multiplier for the rate constraint μ^n has the same function as the sum rate weight ω^n once the Lagrangian is derived and only one version exists for each

user. As such, μ^n can be rewritten as ω^n for notational simplicity.

The next step in creating the SCALE algorithm is to isolate $e^{\tilde{s}_k^n}$ in 2.13 then undoing the change in variable from s_k^n to $e^{\tilde{s}_k^n}$ in order to produce the following iterative PSD update formula:

$$s_k^n = \frac{\omega^n \alpha_k^n}{\lambda^n + \sum_{\substack{m \in \mathcal{N} \\ m \neq n}} \frac{\omega^m H_k^{mn} \alpha_k^m}{\text{int}_k^m}}. \tag{2.14}$$

Furthermore, by including the non-negative PSD and the PSD mask constraints, the PSD update formula then becomes

$$s_k^n = \left[\frac{\omega^n \alpha_k^n}{\lambda^n + \sum_{\substack{m \in \mathcal{N} \\ m \neq n}} \frac{\omega^m H_k^{mn} \alpha_k^m}{\text{int}_k^m}} \right]_k^{s_k^{n,\text{mask}}}. \tag{2.15}$$

Note that β_k^n is not used in the PSD update formula thus only α_k^n needs to be calculated using

$$\begin{aligned} \alpha_k^n &= \frac{\text{SINR}_k^n}{1 + \text{SINR}_k^n} \\ &= \frac{H_k^{nn} s_k^n}{\Gamma \text{rec}_k^n} \end{aligned}$$

where SINR_k^n is the Signal-to-Interference-plus-Noise ratio (SINR) for user n on frequency tone k. The PSD update formula works by finding λ^n such that the total transmit power constraint becomes tight. If it results in a negative λ^n, then λ^n is set to zero. Then, the ω^n for $n \in \mathcal{FM}$ is set such that the FM users rate target constraint is tight.

In the SCALE algorithm process, each user iteratively updates its approximation by recalculating α_k^n then applies the PSD update formula. However, between every iteration, each user needs to know all of its outgoing crosstalk gains and the interference measured by every other user. Hence, a message passing system is required. This message passing can be done through an SMC that has full channel crosstalk gain knowledge. At the end of every iteration, each user sends its measured interference and its PSD to the SMC which combines the necessary elements needed by each user and relays it back to each user before the next PSD update.

Since the SCALE algorithm works over a continuous PSD, a discretized version of SCALE was proposed in [34] where the bit-loadings are constrained to integer units. In

the discretized version, the continuous SCALE is performed then rounded down such that the resulting bit-loadings are integers. Afterwards, the algorithm performs unit-bit-loading increase over the more efficient frequency tones until it reaches the total transmit power constraint.

Distributed Spectrum Balancing

Distributed Spectrum Balancing (DSB) is a locally-optimal algorithm that was first derived in [35] and further developed in [20]. Unlike SCALE where the proper maximum over a lower-bound concave set is found, DSB attempts to solve the RA problem by meeting the KKT conditions. Although the KKT conditions are only sufficient where all locally-optimal solutions meet it, it does not guarantee the other direction where all solutions meeting it are not necessarily locally-optimal because the RA problem due to the difference of convex nature of the RA problem. However, results have shown that it often gives locally-optimal solutions.

The DSB algorithm works by finding a solution to (2.12) that meets the KKT conditions in hopes that it is a locally-optimal solution. Instead of the RA case given in [35, 20] the derivation for the mixed RA-FM case is provided here.

The Lagrangian to the mixed RA-FM problem is

$$\mathcal{L} = \sum_{n\in\mathcal{RA}} \omega^n R^n - \sum_{n\in\mathcal{N}} \lambda^n \Big(\sum_{k\in\mathcal{K}} s_k^n - P^n/\Delta f\Big) - \sum_{n\in\mathcal{FM}} \mu^n (R^{n,\text{target}} - R^n).$$

Taking its derivative over s_k^n gives

$$\frac{\partial \mathcal{L}}{\partial s_k^n} = \lambda^n + \sum_{\substack{m\in\mathcal{RA}\\ m\neq n}} \frac{\omega^m H_k^{mn}}{\ln 2}\left(\frac{1}{\text{int}_k^m} - \frac{1}{\text{rec}_k^n}\right) + \sum_{\substack{m\in\mathcal{FM}\\ m\neq n}} \frac{\mu^m H_k^{mn}}{\ln 2}\left(\frac{1}{\text{int}_k^m} - \frac{1}{\text{rec}_k^n}\right)$$
$$+ \begin{cases} \frac{\omega^n/\ln 2}{\frac{\text{int}_k^n}{H_k^{nn}/\Gamma} + \underbrace{s_k^n}_{*}} & \text{if } n\in\mathcal{RA} \\ \frac{\mu^n/\ln 2}{\frac{\text{int}_k^n}{H_k^{nn}/\Gamma} + \underbrace{s_k^n}_{*}} & \text{if } n\in\mathcal{FM} \end{cases}.$$

As in the SCALE case, the Lagrange multiplier μ^n for the rate constraint plays the same role as the weight ω^n in the sum rate and they are both mutually exclusive on a per-user

basis. Thus, in order to simplify the notation, ω^n will replace μ^n for the FM users. The next step involves isolating the s_k^n denoted by an *. This produces the following PSD update formula:

$$s_k^n = \frac{\omega^n}{\lambda^n + \sum_{\substack{m\in\mathcal{N}\\ m\neq n}} \omega^m H_k^{mn}\left(\frac{1}{\text{int}_k^n} - \frac{1}{\text{rec}_k^n}\right)} - \frac{\Gamma\text{int}_k^n}{H_k^{nn}}$$

where the Lagrange multiplier λ^n includes a normalization by $\ln 2$. With the PSD mask constraints, the PSD update formula becomes

$$s_k^n = \left[\frac{\omega^n}{\lambda^n + \sum_{\substack{m\in\mathcal{N}\\ m\neq n}} \omega^m H_k^{mn}\left(\frac{1}{\text{int}_k^n} - \frac{1}{\text{rec}_k^n}\right)} - \frac{\Gamma\text{int}_k^n}{H_k^{nn}}\right]_0^{s_k^{n,\text{mask}}} . \tag{2.16}$$

Like with SCALE, the PSD update formula is iteratively applied where at every step, λ^n is found such that the total transmit power constraint is tight, or, if that results in a negative λ^n, then it is set to zero. The next step is to adjust ω^n for the FM users by setting it such that their target rate constraint it tight. Also like SCALE, DSB requires a message passing system through an SMC in order to pass around the measured interference, int_k^n, and receive signal power, rec_k^n, between the users.

2.6 Conclusion

This chapter introduced the DSL environment, its challenges, the DSL resource allocation problem and a host of algorithms that were designed for the DSL resource allocation problem. The DSL resource allocation algorithms have a tradeoff between performance and complexity where the simpler algorithms such as PBO and Noise Reference are very easy to implement but are not able to use the DSL channel to its full potential. On the other extreme, the network-level algorithms can produce optimal or near-optimal performances but they require full channel knowledge, have a very high computational complexity in the case of globally-optimal algorithms, and require a full message passing backbone in the case of SCALE and DSB. In the middle, the user-level algorithms can be fully distributive and, at every iteration, only need to measure the interference and knowledge of the direct channel gains. However, their performance is often sub-optimal. Another tradeoff also exists in the form of network information required (such as channel gains and user PSD) and

achievable performance. This is observed with the network-level algorithms requiring full network information in order to produce near-optimal results. Similarly, ASB can perform better than IWF with the use of a reference line that provides some information about the network.

Chapter 3

Generalized Recursive Water-Filling (GRWF)

As discussed in Chapter 2, Iterative Water-Filling (IWF) can be summarized as having each user continuously performing Water-Filling (WF) until an equilibrium is reached. As such, improving the runtime of WF will improve the overall runtime of IWF. The WF procedure described in Chapter 2 uses bisection to find the water-fill level, a costly approach because it converges linearly to the solution. Thus, the desired precision in the solution becomes tightly linked to the number of bisection steps required. This chapter introduces a WF algorithm that can remove that link and make the number of steps completely dependent on the number of frequency tones while still having a smaller computational footprint.

The novel WF algorithm developed in this chapter is based on a recursive approach that allows it to find the water-filling level within a fixed number of steps using maximum total transmit power and minimum target rate constraints concurrently. The algorithm is furthermore generalized for the inclusion of per-tone weights on the bit-loading and on the sum power which enables its use over frequency tones with different bandwidths. A second less powerful novel projection method for solving the water-filling problem is also introduced. For water-filling-like algorithms such as DSB and SCALE, this chapter shows that a recursive solution similar to GRWF is not possible and thus introduces a bit-loading method for water-filling-like algorithms.

This chapter begins with the introduction of one previously explored method and one new method for performing WF in Section 3.1. Following that, Section 3.2 generalizes WF

problem and derives its solution. Using the generalized solution, a new recursive approach is developed in Section 3.3 and used to form GRWF algorithm in Section 3.4. Section 3.5 then compares the GRWF to the previously explored algorithms. Finally, Section 3.6 shows that this recursive formulation cannot be applied to DSB or SCALE even if their update formula is similar to the WF one.

Two notational changes are made in this chapter only. The first change uses d_k^n to denote the power transfer gain to interference plus noise ratio $H_k^{nn}/(\Gamma(\text{int}_k^n + \sigma_k^n))$. The second change omits the n notation to simplify the equations as WF is a single-user algorithm. Thus, for this chapter only, the bit-loading on tone k is now calculated using

$$b_k = \log_2 (1 + s_k d_k)$$

and the WF update formula (2.8) becomes

$$s_k = \left[a - \frac{1}{d_k} \right]_0^{s_k^{\text{mask}}}$$

where a is the WF level.

3.1 Other WF Methods

Aside from bisection, other methods for performing water-filling were developed prior to GRWF and two are presented here: the discrete bit-loading [36, 37] and the projection approach. The former relies on iteratively incrementing the bit-loading by fixed values on the frequency tone that requires the least power for the increment. The projection approach transforms the WF problem into an orthogonal projection problem then into a least-square optimization problem.

3.1.1 Discrete Bit-Loading for WF

The discrete bit-loading approach is an iterative solution to water-filling where, at each step, we increment the bit-loading on the frequency tone that requires the least power. The action performed at each iteration for a constant bit-loading increment B can be

summarized as

$$\begin{aligned} b_k &= b_k + B \\ \text{with} & \\ k &= \underset{k \in \mathcal{K}}{\operatorname{argmin}} \quad s_k(b_k + B) - s_k(b_k) \end{aligned}$$

where b_k is the bit-loading on tone k, and $s_k(b_k)$ is the PSD required for achieving b_k.

This approach heavily relies on the change in PSD required for incrementing the bit-loading. It is thus useful to define the change as a function of the current bit-loading as follows:

$$\begin{aligned} \Delta s_k(b_k) &= s_k(b_k + B) - s_k(b_k) \\ &= \frac{1}{d_k}(2^{b_k+B} - 1) - \frac{1}{d_k}(2^{b_k} - 1) \\ &= \frac{2^B - 1}{d_k} 2^{b_k}. \end{aligned}$$

It is at this point where we see an advantage for using the discrete bit-loading approach. The change in PSD $\Delta s_k(b_k)$ is simple to calculate. Moreover, updating the change in PSD when the frequency tone increments its bit-loading by B is as simple as doing

$$\Delta s_k(b_k + B) = \Delta s_k(b_k)\, 2^B.$$

Algorithm 3.1 summarizes the discrete bit-loading approach using the constant bit-loading increment B. In the algorithm, P^{required} is the total transmit power required, and R is the achievable rate obtained with the generated solution. The first *if* condition in Algorithm 3.1 relates to the total transmit power constraint and stops the algorithm when the rate can no longer be increased without violating the constraint. The second *if* removes the index k from the set $\mathcal{K}$ if the frequency tone k's bit-loading can no longer be incremented without violating the PSD mask constraint. If neither of the constraints are violated, then the PSD, bit-loading, total transmit power required, and rate obtained are updated.

An illustrative example using Algorithm 3.1 is shown in Fig. 3.1. The example shows the order in which the frequency tones are updated. The water-fill floor represents the $1/d_k$ terms for different frequency tones. Notice how at each iteration, the frequency tone

Algorithm 3.1 Discrete Bit-Loading for Water-Filling with Constant Bit-Loading Increments

$\mathcal{K} \leftarrow \{1, 2, \ldots, K\}$
$b_k \leftarrow 0 \quad \forall k \in \mathcal{K}$
$s_k \leftarrow 0 \quad \forall k \in \mathcal{K}$
$P^{\text{used}} \leftarrow 0$
$R \leftarrow 0$
loop
  $k \leftarrow \underset{k \in \mathcal{K}}{\text{argmin}}\ \Delta s_k(b_k)$
  if $P^{\text{used}} + \Delta f\ \Delta s_k(b_k) \geq P$ **then**
    break
  else if $s_k + \Delta s_k(b_k) \geq s_k^{\text{mask}}$ **then**
    $\mathcal{K} \leftarrow \mathcal{K} \setminus k$
  else
    $s_k \leftarrow s_k + \Delta s_k(b_k)$
    $b_k \leftarrow b_k + B$
    $P^{\text{used}} \leftarrow P^{\text{used}} + \Delta f\ \Delta s_k(b_k)$
    $R \leftarrow R + f_s B$
  end if
end loop

with the lowest floor gets updated. In fact, the frequency tone with the lowest floor always requires the smallest change in PSD $\Delta s_k(b_k)$.

The example in Fig. 3.1 shows the discrete bit-loading approach's disadvantage. At every iteration, the approach must find the frequency tone with the lowest floor, which is the same as finding the one requiring the smallest $\Delta s_k(b_k)$ for a bit-loading increment. Thus, the algorithm must search through K variables at every iteration. Moreover, the number of iteration can be relatively high because it is directly linked to the achievable bit rate. Assuming a 20 Mb/s sum rate, and a unit bit-loading increment, Algorithm 3.1 would require 5000 iterations and will therefore need to perform 5000 searches through K variables. One remedy is to produce every possible $\Delta s_k(b_k)$ and sorting them before entering the loop. This will effectively remove the argmin step in the algorithm in exchange of a sorting process.

The discrete bit-loading approach presented here uses a constant bit-loading increment. A non-constant increment is also possible and useful for cases when only a few bit-loading values are allowed, e.g., {1,2,4,8} b/s/Hz. In this scenario, instead of choosing the frequency

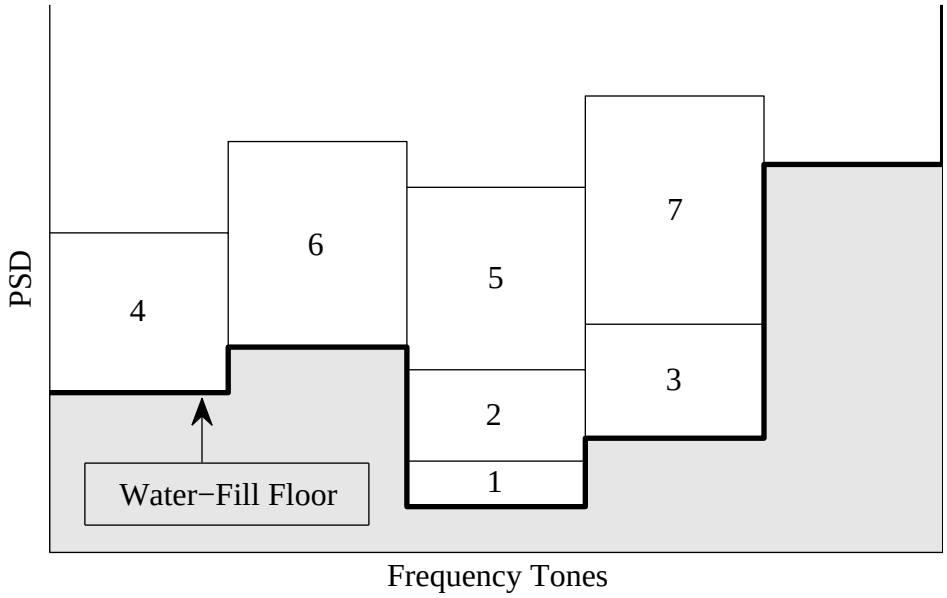


Fig. 3.1 Example of the discrete bit-loading method showing the order it assigns the bit-loading increments.

tone that requires the least power to increment its bit-loading value, the algorithm chooses the frequency tone that offers the smallest power-increase-to bit-loading-increment ratio. The constant bit-loading increment is actually a special case that can ignore the now-constant denominator in the ratio.

3.1.2 Projection for WF

Unlike the discrete bit-loading approach, the projection approach produces a solution in a continuous space. This approach starts by noticing that the total transmit power constraint inequality is tight at the WF solution. This is due to the bit-loading being a monotonic function. That is, an increase in power always results in an increase in the bit-loading. Thus, the maximum achievable rate will require the maximum total transmit power. Using this monotonic property, the WF level can be obtained by directly solving for it in the total

transmit power constraint as follows:

$$\begin{aligned}\frac{P}{\Delta f} &= \sum_{k\in\mathcal{K}} s_k \\ &= Ka - \sum_{k\in\mathcal{K}} \frac{1}{d_k}. \end{aligned} \tag{3.1}$$

With (3.1), the water-filling level can be found in one step then the PSD can be generated in another step. Actually, (3.1) only holds when all K frequency tones have non-zero PSD and when the PSD mask constraints are met, and this will be addressed shortly. Nevertheless, solving for a in (3.1) can be reformulated into a orthogonal projection problem. Under this reformulation, the point

$$\left(-\frac{1}{d_k}, -\frac{1}{d_2}, \ldots, -\frac{1}{d_K}\right) \tag{3.2}$$

would be orthogonally projected onto the maximum total transmit power hyperplane defined as

$$s_1 + s_2 + \ldots + s_K = \frac{P}{\Delta f}.$$

The orthogonal projection view spawns from observing that s_k is a linear function of a. Therefore, varying a^n generates a straight line in K-dimensions that passes through the point (3.2). That line has a slope of $(1, 1, \ldots, 1)$ and is therefore normal to the maximum total transmit power hyperplane.

Obviously, this simplistic case does not apply to most scenarios because it is possible that some frequency tones will have negative PSD using orthogonal projection. Additionally, the algorithm must respect the PSD masks. As such, we bound the projection hyper-plane using the non-zero PSD and the PSD mask constraints to finally arrive with the following bounded hyper-plane:

$$s_1 + s_2 + \ldots + s_K = \frac{P}{\Delta f}$$

such that

$$0 \leq s_k \leq s_k^{\text{mask}} \quad \forall k \in \mathcal{K}.$$

The question now is how to project the point onto the bounded hyper-plane. Two cases can arise and both are presented in Fig. 3.2 in a two-tone example. The first case starts

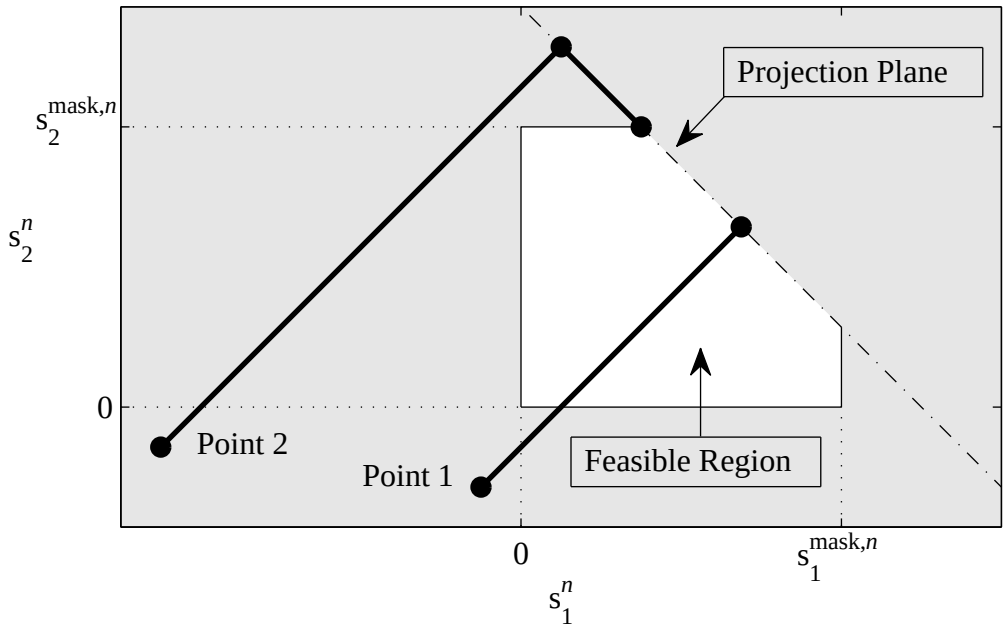


Fig. 3.2 Water-filling projection example on a 2-dimensional plane using multiple starting points.

with Point 1. Here, the point can be orthogonally projected onto the plane and still be within the feasible region. As such, no other step needs to be done and the projection can be used as the water-filling solution. The second case starts with Point 2. In this case, the point's orthogonal projection lies outside the feasible region. The next step is to find the nearest point on the bounded plane to the projection.

Actually, both cases can be summarized as finding the point on the bounded plane that is nearest to the point (3.2). Thus, this becomes a minimum distance problem and the bounded orthogonal projection approach can be rewritten as the following least-squares optimization problem:

$$\begin{aligned} \min_{s_k^n \forall k \in \mathcal{K}} \quad & \sum_{k \in \mathcal{K}} \left(s_k + \frac{1}{d_1} \right)^2 \\ \text{such that} \quad & \frac{P}{\Delta f} = \sum_{k \in \mathcal{K}} s_k \\ & 0 \leq s_k \leq s_k^{\text{mask}} \quad \forall k \in \mathcal{K}. \end{aligned}$$

Algorithm 3.2 Projection Approach to Unbounded Water-Filling

$\mathcal{K} \leftarrow \{1, 2, \ldots, K\}$
repeat
  $a \leftarrow \frac{P/\Delta f + \sum_{k\in\mathcal{K}} 1/d_k}{\sum_{k\in\mathcal{K}} 1}$
  $s_k \leftarrow a - \frac{1}{d_k}, \quad \forall k \in \mathcal{K}$
  for $k \in \mathcal{K}$ **do**
    if $s_k \leq 0$ **then**
      $s_k \leftarrow 0$
      $\mathcal{K} \leftarrow \mathcal{K} \setminus k$
    end if
  end for
until convergence

Needless to say, this approach has to solve a constrained quadratic programming and is thus more complex and likely more time-consuming than the bisection method. Nevertheless, when no PSD mask is enforced, this projection approach can be simplified into Algorithm 3.2. The algorithm attempts to the water-filling level a by solving for it in (3.1). After solving for a, it checks for consistency by checking for negative PSD values. If those values exist, it removes the frequency tones associated to them and attempts to solve for a again until the PSD is feasible.

With no PSD mask enforced, the discrete bit-loading approach is no match to the projection approach. With an enforced PSD mask, the opposite holds since the discrete bit-loading's algorithm does not really change and the projection approach will require quadratic programming. Even though both algorithms can offer an improvement on bisection, the discrete bit-loading approach has a number of iteration that grows with the desired rate and the projection approach can result in trying to solve a more complex optimization problem.

3.2 Generalized Water-Filling

Three methods have been presented up to this point: bisection, which converges linearly to the solution; discrete bit-loading, which finds the discrete solution to water-filling in a finite number of steps, but that number of steps can be potentially large; and projection, which relies on least-squares optimization once PSD mask is involved. Before introducing

the generalized recursive approach to water-filling, this section will present the generalized water-filling problem, its solution, and a short analysis on its solution.

The water-filling problem used in the three methods did not consider per-tone weights. Weights on the bit-loading b_k are useful for cases where one might want to penalize or favorize the bit-loading on certain frequency tones or use different symbol rates for each frequency tone. Similarly, one may want to penalize or favorize the power usage on certain frequency tones or use different symbol spacing between different frequency tones.

To include the weights, the single-user version of the Rate Adaptive given by (2.3) is modified into

$$\begin{aligned} \max_{s_k \forall k \in \mathcal{K}} \quad & \sum_{k \in \mathcal{K}} \omega_k \log_2 (1 + s_k d_k) \\ \text{such that} \quad & \sum_{k \in \mathcal{K}} \chi_k s_k \leq P \\ & 0 \leq s_k \leq s_k^{\text{mask}}, \quad \forall k \in \mathcal{K} \end{aligned}$$

where d_k represents the $\frac{1}{\Gamma} \frac{H_k^n n}{\text{int}_k^n}$ term, ω_k^n is the capacity's weighting factor, and χ_k^n the total transmit power weighting factor. The symbol rate f_s and DMT tone spacing Δf are omitted since they can be absorbed by the weighting factors. Likely, to include the weights, the single-user version of the Fixed Margin problem (2.4) is modified into

$$\begin{aligned} \min_{s_k \forall k \in \mathcal{K}} \quad & \sum_{k \in \mathcal{K}} \frac{s_k}{\omega_k} \\ \text{such that} \quad & \sum_{k \in \mathcal{K}} \frac{\log_2 (1 + s_k d_k)}{\chi_k} \geq R \\ & 0 \leq s_k \leq s_k^{\text{mask}}, \quad \forall k \in \mathcal{K}. \end{aligned}$$

Note that the same weighting factors are used in both RA and FM problems but in different ways. This will be addressed later.

The following two theorems show the solutions to the RA and FM problems.

Theorem 1. *The solution to the generalized RA problem is*

$$s_k = \left[a \frac{\omega_k}{\chi_k} - \frac{1}{d_k} \right]_0^{s_k^{\text{mask}}} \tag{3.3}$$

where the variable a is chosen such that the total transmit power constraint inequality is tight.

Proof. The RA adaptive problem can be solved using its Lagrange function as it was the case for the non-generalized water-filling problem. The following shows the Lagrange function where λ is the dual variable associated to total power constraint, ν_k to the non-negative power constraint, and μ_k to the spectral mask.

$$L_{\mathrm{RA}} = \sum_{k\in\mathcal{K}} \omega_k \log_2\left(1 + s_k d_k\right) - \lambda\left(\sum_{k\in\mathcal{K}} \chi_k s_k - P\right) + \sum_{k\in\mathcal{K}} \left(\nu_k s_k - \mu_k (s_k - s_k^{\mathrm{mask}})\right)$$

The Lagrange function's saddle point is given by

$$0 = \frac{\partial L_{\mathrm{RA}}}{\partial s_k} = \frac{\omega_k d_k}{(1 + s_k d_k)\ln 2} - \lambda\chi_k + \nu_k - \mu_k$$

and solving it produces the following update formula

$$s_k = \frac{\omega_k}{(\lambda\chi_k - \nu_k + \mu_k)\ln 2} - \frac{1}{d_k} \tag{3.4}$$

where λ is chosen such that the total power constraint inequality is tight, ν_k and μ_k are non-negative values chosen to keep s_k within the box constraints when necessary. Thus, ν_k and μ_k can be omitted by performing the following equivalent update formula:

$$s_k = \left[a\frac{\omega_k}{\chi_k} - \frac{1}{d_k}\right]_0^{s_k^{\mathrm{mask}}}$$

where a is chosen such that the total transmit power constraint is tight and $[\cdot]_x^y$ is a box constraint function. □

Theorem 2. *The solution to the FM problem is*

$$s_k = \left[a\frac{\omega_k}{\chi_k} - \frac{1}{d_k}\right]_0^{s_k^{\mathrm{mask}}}$$

where the variable a is chosen such that the target rate constraint inequality is tight.

Proof. Like the generalized RA problem, the generalized FM problem can also be solved

with its Lagrange function. The following shows the Lagrange function where λ is the dual variable associated to target rate constraint, ν_k to the non-negative power constraint, and μ_k to the spectral mask.

$$L_{\mathrm{FM}} = -\sum_{k\in\mathcal{K}} \frac{s_k}{\omega_k} - \lambda\left(R - \sum_{k\in\mathcal{K}} \frac{1}{\chi_k}\log_2\left(1+s_k d_k\right)\right) + \sum_{k\in\mathcal{K}} \left(\nu_k s_k - \mu_k(s_k - s_k^{\mathrm{mask}})\right)$$

Similar to the generalized RA problem, the saddle point for the FM Lagrange function is given by

$$0 = -\frac{1}{\omega_k} + \frac{\frac{\lambda}{\chi_k} d_k}{(1+s_k d_k)\ln 2} + \nu_k - \mu_k.$$

This produces a solution slightly different to (3.4) in terms of Lagrange variable placements

$$s_k = \frac{\frac{\lambda}{\chi_k}}{(\frac{1}{\omega_k} - \nu_k + \mu_k)\ln 2} - \frac{1}{d_k}$$

where λ is chosen such that the target rate constraint inequality is tight, ν_k and μ_k are non-negative values chosen to keep s_k within the box constraints when necessary. That procedure is equivalent to doing

$$s_k = \left[a\frac{\omega_k}{\chi_k} - \frac{1}{d_k}\right]_0^{s_k^{\mathrm{mask}}}$$

where a is chosen such that the target rate constraint is tight. □

Theorems 1 and 2 show an interesting result: both generalized RA and RM problems use the same approach to arrive to their solutions. Both apply (3.3) and select a such that their total transmit power or target rate constraint becomes active. Moreover, the theorems also show a link between the weighting factors used in both problems. If χ_k is removed by setting it to one, then it is easy to see that the weights ω_k in the RA objective function have an inversely proportional effect on weighted sum in the FM problem's objective function. The same analysis can be done on χ_k by removing ω_k.

Another interesting aspect is that only the ω_k/χ_k ratio matters in the update formula (3.3). Thus, if both RA and FM problems use the same bit-loading weights and the same

transmit power weights such that both use

$$R = \sum_{k \in \mathcal{K}} \omega_k \log_2 (1 + s_k d_k)$$

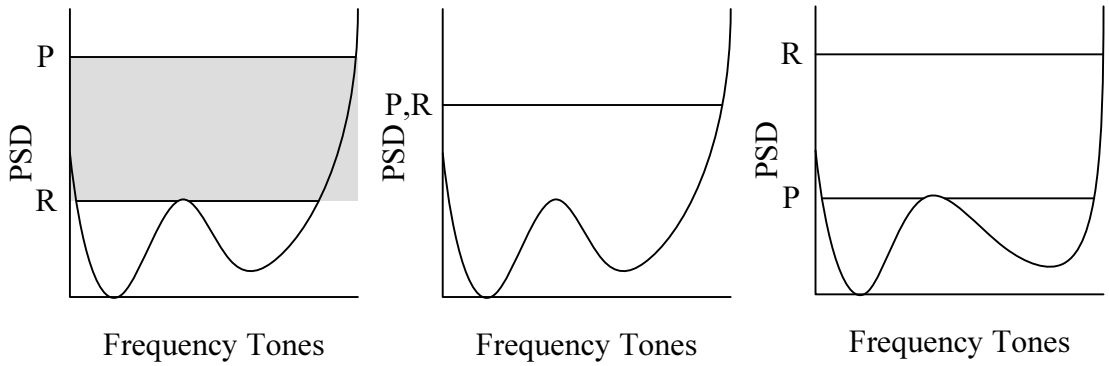
and

$$P = \sum_{k \in \mathcal{K}} \chi_k s_k,$$

the ratio would be maintained. In this special case, the only difference between the RA and FM problem will be the different water-fill level. Three cases can arise and are represented in Fig. 3.3. In the first case in Fig. 3.3a, the water-fill level generated by the RA problem is higher than that of the FM problem. In this case, the two levels form a feasible region shown in grey in which all water-fill levels meet the target rate and the total transmit power constraints. The second case in 3.3b has both RA and FM water-fill levels at the same height thus limiting the feasible region to only one solution. Finally, the third case in 3.3c has the RA water-fill level below the FM one thus there is no feasible solution.

Using both total transmit power and target rate constraints conjointly is at the center of the power control algorithm in [7]. With the power control algorithm in [7], the modified IWF algorithm continuously adjusts each user's total transmit power such that each user's rate is equal to their target rates. However, instead of using both constraints simultaneously, the algorithm performs water-filling on each user using the total transmit power constraint only, then, between each water-filling iteration, it increments or decrements the total transmit power depending on the obtained rates. However, the total transmit power value was made to never exceed a certain value, the true maximum total transmit power. This process is wasteful as many iterations can be needed before finding the right total transmit power limits. With the results from Theorems 1 and 2, we know that both constraints can be used simultaneously. To obtain the same results as the power control algorithm in [7], the water-filling procedure water-fills until the first constraint is met. That is, we water-fill until the target rate constraint is met only if the total transmit power constraint remains unviolated. Otherwise, we water-fill until the total transmit power constraint is met. Selective Iterative Water-filling (SIW) [38, 39] is another algorithm that requires the use of both constraints.

A workaround is to form a hard bound system. Whichever constraint acts as the hard bound will also have priority when there is no feasible region. For example, a hard bound



(a) Case 1 (b) Case 2 (c) Case 3

Fig. 3.3 Water-Filling using desired rate and total transmit power constraints concurrently. R denotes the water-fill level required by the target rate constraint, and P denotes the highest water-fill level allowed by the total transmit power constraint. The grey area is the feasible area where any water-fill level inside will meet the both constraints.

on the total transmit power will guarantee that the transmit power will never exceed a threshold at the penalty of not always being able to achieve the desired rate.

It is also interesting to note that at high transmit power with no spectral mask, the relationship between each frequency tone's PSD will have the same relationship as the ratios $\frac{\omega_k}{\chi_k}$. Thus, given two frequency tones with one having a weight ω_k twice as big as the other one, the frequency tone with the larger weight will roughly have a PSD twice as high as the other one at high transmit power.

3.3 Recursive Formulation

The recursive formulation works by trying a finite number of chosen water-filling levels. The ones of interest are the ones where at s_k transitions from inactive to active or from active to saturated at s_k^{mask}. In other words, we are interested in the smallest water-fill level that will make s_k non-zero and the largest water-fill level that will keep s_k below s_k^{mask}. Thus, the values are interest are given by

$$a_k^0 = \frac{\chi_k}{\omega_k d_k}, \tag{3.5}$$

which corresponds to the water-fill level where s_k transitions from inactive to active, and by

$$a_k^{\text{mask}} = \frac{\chi_k s_k^{\text{mask}}}{\omega_k} + \frac{\chi_k}{\omega_k d_k} = \frac{\chi_k s_k^{\text{mask}}}{\omega_k} + a_k^0, \tag{3.6}$$

which corresponds to the water-fill level where s_k transitions from active to saturated.

The first step is to produce all of the aforementioned water-fill levels of interest. The second step is to order the water-fill levels in ascending order. Let a_i^{all} be ordered and contain every element from a_k^0 and a_k^{mask}, i.e., a_i^{all} now contains $2K$ ordered elements. Let the function $k = f(i)$ denote the function that will map i back to the presorted index k. That is, if a_4^0 becomes a_7^{all} after the combination and ordering process, then $f(7)$ produces 4. Moreover, let $\mathcal{N}_i^0 \triangleq \{k \in \mathcal{K} : a_k^0 \leq a_i^{\text{all}}, f^{-1}(k) \leq i\}$ and $\mathcal{N}_i^{\text{mask}} \triangleq \{k \in \mathcal{K} : a_k^{\text{mask}} \leq a_i^{\text{all}}, f^{-1}(k) \leq i\}$. With these two sets, we have

$$\{a_k^0 \quad \forall k \in \mathcal{N}_i^0\} \cup \{a_k^{\text{mask}} \quad \forall k \in \mathcal{N}_i^{\text{mask}}\} = \{a_k^{\text{all}} : 1 \leq k \leq i, k \in \mathbb{Z}\}.$$

When i increments by 1 until it reaches $2K$, either $\mathcal{N}_i^0$ or $\mathcal{N}_i^{\text{mask}}$ grows by 1 element. Additionally, $\mathcal{N}_i^{\text{mask}} \subseteq \mathcal{N}_i^0$ always holds.

The second step is to find the required power and the obtained rate associated to each water-fill level in $\mathcal{A}^*$ which are denoted by P_i^* and R_i^*, respectively, through the use of recursion. That is, P_i^* and R_i^* should be a function of a_i^{all}, a_{i-1}^{all}, P_{i-1}^*, and R_{i-1}^* only.

Before going any further, we will establish that P_1^* and R_1^* are both zero because the water-fill level they are associated to, a_1^{all}, is actually the smallest a_k^0. That is, below this water-fill level, all s_k are inactive. For further use, we shall also define a new variable W_i obtained with the following recursive equation:

$$\begin{aligned} W_i &= \sum_{k \in \mathcal{N}_i^0 \setminus \mathcal{N}_i^{\text{mask}}} \omega_k \\ &= \sum_{k \in \mathcal{N}_i^0} \omega_k - \sum_{k \in \mathcal{N}_i^{\text{mask}}} \omega_k \\ &= W_{i-1} + \begin{cases} \omega_{f(i)} & \text{if } \mathcal{N}_i^{\text{mask}} = \mathcal{N}_{i-1}^{\text{mask}} \\ -\omega_{f(i)} & \text{if } \mathcal{N}_i^0 = \mathcal{N}_{i-1}^0 \end{cases} \end{aligned}$$

with W_0 initialized at 0.

The following propositions show the recursive formula needed to obtain the other values of P_i^* and R_i^* using the RA notation for the weighting factors.

Proposition 1. *For indices greater than 1, the recursive formula for calculating the required power given the water-fill level a_i^{all} is*

$$P_i^* = P_{i-1}^* + (a_i^{\text{all}} - a_{i-1}^{\text{all}})W_{i-1}. \tag{3.7}$$

Proof. We begin with the weighted power sum

$$P_i^* = \sum_{k\in\mathcal{K}} \chi_k s_k$$

then we replace it with its equivalent form using the sets $\mathcal{N}_i^0$ and $\mathcal{N}_i^{\text{mask}}$ and substitute s_k with (3.3) and s_k^{mask} where appropriate

$$P_i^* = \sum_{k\in\mathcal{N}_i^0\setminus\mathcal{N}_i^{\text{mask}}} \chi_k \left(a_i^{\text{all}} \frac{\omega_k}{\chi_k} - \frac{1}{d_k} \right) + \sum_{k\in\mathcal{N}_i^{\text{mask}}} \chi_k s_k^{\text{mask}}.$$

The first summation sums over the active and unsaturated s_k and the second summation sums over the saturated s_k. To obtain the recursive form, we subtract both sides by P_{i-1}^* as follows

$$\begin{aligned} P_i^* - P_{i-1}^* = &\sum_{k\in\mathcal{N}_i^0\setminus\mathcal{N}_i^{\text{mask}}} \left(a_i^{\text{all}}\omega_k - \frac{\chi_k}{d_k} \right) + \sum_{k\in\mathcal{N}_i^{\text{mask}}} \chi_k s_k^{\text{mask}} \\ &- \sum_{k\in\mathcal{N}_{i-1}^0\setminus\mathcal{N}_{i-1}^{\text{mask}}} \left(a_{i-1}^{\text{all}}\omega_k - \frac{\chi_k}{d_k} \right) - \sum_{k\in\mathcal{N}_{i-1}^{\text{mask}}} \chi_k s_k^{\text{mask}}. \end{aligned}$$

After some simplifications, we obtain

$$P_i^* - P_{i-1}^* = (a_i^{\text{all}} - a_{i-1}^{\text{all}})W_{i-1} + \begin{cases} a_i^{\text{all}}\omega_{f(i)} - \frac{\chi_{f(i)}}{d_{f(i)}} & \text{if } \mathcal{N}_i^{\text{mask}} = \mathcal{N}_{i-1}^{\text{mask}} \\ \chi_{f(i)} s_{f(i)}^{\text{mask}} - a_i^{\text{all}}\omega_{f(i)} - \frac{\chi_{f(i)}}{d_{f(i)}} & \text{if } \mathcal{N}_i^0 = \mathcal{N}_{i-1}^0 \end{cases}. \tag{3.8}$$

When $\mathcal{N}_i^{\text{mask}} = \mathcal{N}_{i-1}^{\text{mask}}$, a_i^{all} is actually $a_{f(i)}^0$ and $a_{f(i)}^0\omega_{f(i)} - \frac{\chi_{f(i)}}{d_{f(i)}}$ becomes 0. Similarly, when $\mathcal{N}_i^0 = \mathcal{N}_{i-1}^0$, a_i^{all} takes its value from $a_{f(i)}^{\text{mask}}$ and $\chi_{f(i)}s_{f(i)}^{\text{mask}} - a_{f(i)}^{\text{mask}}\omega_{f(i)} - \frac{\chi_{f(i)}}{d_{f(i)}}$ becomes 0.

Thus, the final term in (3.8) can be removed and the recursive formula becomes

$$P_i^* = P_{i-1}^* + (a_i^{\text{all}} - a_{i-1}^{\text{all}})W_{i-1}.$$

□

Proposition 2. *For indices greater than 1, the recursive formula for calculating the achieved rate given the WF level a_i^{all} is*

$$\begin{aligned}\left(R_i^* - \log_2(a_i^{\text{all}})W_{i-1}\right) &= \left(R_{i-1}^* - \log_2(a_{i-1}^{\text{all}})W_{i-2}\right) \\ &+ \begin{cases} -\omega_{f(i)} \log_2 a_{i-1}^{\text{all}} & \text{if } \mathcal{N}_i^{\text{mask}} = \mathcal{N}_{i-1}^{\text{mask}} \\ \omega_{f(i)} \log_2 a_{i-1}^{\text{all}} & \text{if } \mathcal{N}_i^0 = \mathcal{N}_{i-1}^0 \end{cases}\end{aligned} \tag{3.9}$$

Proof. We begin with the weighted rate sum

$$R_i^* = \sum_{k \in \mathcal{K}} \omega_k \log_2(1 + s_k d_k)$$

then we replace it with its equivalent form using the sets $\mathcal{N}_i^0$ and $\mathcal{N}_i^{\text{mask}}$ and substitute s_k with (3.3) and s_k^{mask} where appropriate

$$R_i^* = \sum_{k \in \mathcal{N}_i^0 \setminus \mathcal{N}_i^{\text{mask}}} \omega_k \log_2\left(\frac{a_i^{\text{all}} \omega_k d_k}{\chi_k}\right) + \sum_{k \in \mathcal{N}_i^{\text{mask}}} \omega_k \log_2(1 + s_k^{\text{mask}} d_k).$$

The first summation shows the contribution from the active and unsaturated s_k while the second summation shows the contribution from the saturated s_k. We then decompose the logarithms as follows

$$\begin{aligned}R_i^* = &\sum_{k \in \mathcal{N}_i^0 \setminus \mathcal{N}_i^{\text{mask}}} \omega_k \log_2(a_i^{\text{all}}) - \sum_{k \in \mathcal{N}_i^0 \setminus \mathcal{N}_i^{\text{mask}}} \omega_k \log_2\left(\frac{\chi_k}{\omega_k d_k}\right) \\ &+ \sum_{k \in \mathcal{N}_i^{\text{mask}}} \omega_k \log_2\left(\frac{\chi_k s_k^{\text{mask}}}{\omega_k} + \frac{\chi_k}{\omega_k d_k}\right) - \sum_{k \in \mathcal{N}_i^{\text{mask}}} \omega_k \log_2\left(\frac{\chi_k}{\omega_k d_k}\right)\end{aligned}$$

and simplify the equation to

$$R_i^* = W_i \log_2\left(a_i^{\text{all}}\right) - \sum_{k\in\mathcal{N}_i^0} \omega_k \log_2\left(\frac{\chi_k}{\omega_k d_k}\right) + \sum_{k\in\mathcal{N}_i^{\text{mask}}} \omega_k \log_2\left(\frac{\chi_k s_k^{\text{mask}}}{\omega_k} + \frac{\chi_k}{\omega_k d_k}\right).$$

The above equation can be rewritten as

$$R_i^* = W_{i-1} \log_2\left(a_i^{\text{all}}\right) - \sum_{k\in\mathcal{N}_{i-1}^0} \omega_k \log_2\left(\frac{\chi_k}{\omega_k d_k}\right) + \sum_{k\in\mathcal{N}_{i-1}^{\text{mask}}} \omega_k \log_2\left(\frac{\chi_k s_k^{\text{mask}}}{\omega_k} + \frac{\chi_k}{\omega_k d_k}\right)$$
$$+ \begin{cases} \omega_{f(i)} \log_2\left(a_i^{\text{all}}\right) - \omega_{f(i)} \log_2\left(\frac{\chi_{f(i)}}{\omega_{f(i)} d_{f(i)}}\right) & \text{if } \mathcal{N}_i^{\text{mask}} = \mathcal{N}_{i-1}^{\text{mask}} \\ -\omega_{f(i)} \log_2\left(a_i^{\text{all}}\right) + \omega_{f(i)} \log_2\left(\frac{\chi_{f(i)} s_{f(i)}^{\text{mask}}}{\omega_{f(i)}} + \frac{\chi_{f(i)}}{\omega_{f(i)} d_{f(i)}}\right) & \text{if } \mathcal{N}_i^0 = \mathcal{N}_{i-1}^0 \end{cases}.$$

In the first case where $\mathcal{N}_i^{\text{mask}} = \mathcal{N}_{i-1}^{\text{mask}}$, a_i^{all} is equivalent to $a_{f(i)}^0$ and the fourth and fifth terms cancel each other out. Similarly, in the second case where $\mathcal{N}_i^0 = \mathcal{N}_{i-1}^0$, a_i^{all} is equivalent to a_i^{mask} and the fourth and fifth terms cancel each other out. Hence, the conditional part always evaluates to 0 and can be omitted.

To obtain the recursive form, we subtract both sides by R_{i-1}^* as follows

$$R_i^* - R_{i-1}^* = W_{i-1} \log_2\left(a_i^{\text{all}}\right) - \sum_{k\in\mathcal{N}_{i-1}^{\min}} \omega_k \log_2\left(\frac{\chi_k}{\omega_k d_k}\right) + \sum_{k\in\mathcal{N}_{i-1}^{\max}} \omega_k \log_2\left(\frac{\chi_k s_k^{\text{mask}}}{\omega_k} + \frac{\chi_k}{\omega_k d_k}\right)$$
$$- W_{i-2} \log_2\left(a_{i-1}^{\text{all}}\right) + \sum_{k\in\mathcal{N}_{i-2}^{\min}} \omega_k \log_2\left(\frac{\chi_k}{\omega_k d_k}\right) - \sum_{k\in\mathcal{N}_{i-2}^{\max}} \omega_k \log_2\left(\frac{\chi_k s_k^{\text{mask}}}{\omega_k} + \frac{\chi_k}{\omega_k d_k}\right).$$

After some simplifications, we obtain

$$\left(R_i^* - \log_2\left(a_i^{\text{all}}\right) W_{i-1}\right) = \left(R_{i-1}^* - \log_2\left(a_{i-1}^{\text{all}}\right) W_{i-2}\right) + \begin{cases} -\omega_{f(i)} \log_2 a_{i-1}^{\text{all}} & \text{if } \mathcal{N}_{i-1}^{\text{mask}} = \mathcal{N}_{i-2}^{\text{mask}} \\ \omega_{f(i)} \log_2 a_{i-1}^{\text{all}} & \text{if } \mathcal{N}_{i-1}^0 = \mathcal{N}_{i-2}^0 \end{cases}.$$

□

Propositions 1 and 2 show how we can obtain the required power and obtained rate over a the set $\mathcal{A}^*$ of water-fill levels using recursion. Actually, (3.7) and (3.9) can calculate the power required and the rate obtained for any water-fill level. It can be shown by expanding Proposition 1 that the power required for a water-fill level a in between two consecutive

elements in $\mathcal{A}^*$ such that $A^*_{i-1} \leq a^* \leq A^*_i$ and $i > 1$ is

$$P^{\text{required}} = P^*_{i-1} + (a - A^*_{i-1})W_{i-1}. \tag{3.10}$$

Similarly, it can be shown by expanding Proposition 2 that the rate obtained for a water-fill level a in between two consecutive elements in $\mathcal{A}^*$ such that $A^*_{i-1} \leq a \leq A^*_i$ and $i > 1$ is

$$R = \left(R^*_{i-1} - \log_2\left(A^*_{i-1}\right)W_{i-2}\right) + \log_2\left(a\right)W_{i-1}. \tag{3.11}$$

This leads to the next step in this recursive formulation where the optimal water-fill level is found. The optimal level can be found by first determining the interval $[A^*_{i-1}, A^*_i)$ that encloses it. Under the RA formulation, we find the smallest index i such that the total transmit power constraint is violated. Under the FM formulation, we find the smallest index i such that the desired rate is met. Both cases can benefit from the recursive method for finding the total transmit power required and the rate obtained. Once the interval is known, (3.10) and (3.11) can be used to solve for the water-filling level as follows

$$a^* = \begin{cases} \frac{P - P^*_{i-1}}{W_{i-1}} + A^*_{i-1} & \text{if RA} \\ \exp\left(\frac{R^{\text{target}} - \left(R^*_{i-1} - \log_2\left(A^*_{i-1}\right)W_{i-2}\right)}{W_{i-1}}\right) & \text{if FM} \end{cases}.$$

With the water-fill level a^* now known, the final step in this recursive approach is to calculate the PSD using (3.3). There is one exception that occurs when the interval does not exist, it means that none of the water-filling levels A^*_i were high enough to make the power or rate constraint inequalities tight. More importantly, it means that the PSD mask constraint is the main limiting factor and that the solution is the PSD mask itself.

3.4 Generalized Recursive Water-Filling Algorithm

The section presents the generalized recursive water-filling algorithm that uses total power, target rate, and power spectral density mask constraints, and variable weights. The algorithm has been designed to perform water-filling until the rate is equal to the target rate as long as the total transmit power constraint remains unviolated. Under the violated case, the algorithm will then water-fill until the total transmit power constraint inequality

is tight. The generalized recursive water-filling algorithm is presented in Algorithm 3.3. The sort used in the algorithm does not need to be a stable sort, that is, the values of the magnitude do no need to preserve their original order.

Since Algorithm 3.3 is a generalized case, special cases can be defined. When the target rate constraint is not required, the target rate can either be set to infinity or one can remove everything related to the variables R_k^*. Similarly, when the total power constraint is not required, the maximum total transmit power can either be set to infinity or one can remove everything related to the variable P_k^*. When the power spectral density mask is not required, the mask value can either be set to infinity or one make the set $\mathcal{A}^{\max}$ empty. In fact, for the algorithm to function, only one of the following must be met:

$$\begin{aligned} & P \neq \infty, \\ \text{or } & R^{\text{target}} \neq \infty, \\ \text{or } & s_k^{\text{mask}} \neq \infty \quad \forall k \in \mathcal{K}. \end{aligned}$$

To better compare the performance of the GRWF approach to the three water-filling methods previously described (bisection, discrete bit-loading, and projection), a special case of the GRWF algorithm is presented in Algorithm 3.4. This special case solves the RA formulation, where the desired rate constraint is removed, and where the weights ω_k are constant at a value of f_s for all k and the weights χ_k are constant at a value of Δf for all k. One interesting note is that W_k can now be interpreted as the number of active frequency tones at step k. By setting the PSD mask to infinity over every frequency tone, Algorithm 3.4 degenerates into the closed-form water-filling solution introduced in [40].

3.5 Computational Load

The computational complexity of all four water-filling algorithms presented are of order $\mathcal{O}(K)$, with the potential exception of the projection approach due to its least-squares optimization requirement. Thus, the GRWF, discrete bit-loading, and bisection approaches have the same computational complexity. The projection approach is more complex and it will not be included in further comparisons. So, in order to better compare the three equally-complex algorithms, this section will look at each of their computational load. The comparison will be performed in two parts. The first will look at the RA formulation with

Algorithm 3.3 Generalized Recursive Water-Filling Algorithm

$i \leftarrow 1, W_0 \leftarrow 0, P_1^* \leftarrow 0, R_1^* \leftarrow 0\, R_1^{\text{base}} \leftarrow 0$
$A_k^{\min} \leftarrow \frac{\chi_k}{\omega_k d_k}, A_k^{\max} \leftarrow \frac{\chi_k s_k^{\text{mask}}}{\omega_k} + \frac{\chi_k}{\omega_k d_k}, \quad \forall k \in \mathcal{K}$
$\mathcal{A}^* \leftarrow \text{sort}(\mathcal{A}^{\min} \cup \mathcal{A}^{\max})$
while $(R_i^* \leq R^{\text{target}}$ and $P_i^* \leq P)$ **do**
  $i \leftarrow i + 1$
  if $i = 2K + 1$ **then**
    return $s_k \leftarrow s_k^{\text{mask}}, \quad \forall k \in \mathcal{K}$
  end if
  if $A_{i-1}^* \in \mathcal{A}^{\min}$ **then**
    $W_{i-1} = W_{i-2} + \omega_{f(i-1)}$
    $R_i^{\text{base}} \leftarrow R_{i-1}^{\text{base}} - \log_2(A_{i-1}^*)$
  else
    $W_{i-1} = W_{i-2} - \omega_{f(i-1)}$
    $R_i^{\text{base}} \leftarrow R_{i-1}^{\text{base}} + \log_2(A_{i-1}^*)$
  end if
  $P_i^* \leftarrow P_{i-1}^* + (A_i^* - A_{i-1}^*)W_{i-1}$
  $R_i^* \leftarrow R_i^{\text{base}} + W_{i-1}\log_2(A_i^*)$
end while
if $(R_i^* \geq R^{\text{target}})$ and $(P_i^* \geq P)$ **then**
  $a^* \leftarrow \min\left(2^{\left(\frac{R^{\text{target}} - R_i^{\text{base}}}{W_{i-1}}\right)}, \frac{P - P_{i-1}^*}{W_{i-1}} + A_{i-1}^*\right)$
else if $(R_i* \geq R^{\text{target}})$ **then**
  $a^* \leftarrow 2^{\left(\frac{R^{\text{target}} - R_i^{\text{base}}}{W_{i-1}}\right)}$
else
  $a^* \leftarrow \frac{P - P_{i-1}^*}{W_{i-1}} + A_{i-1}^*$
end if
return $s_k \leftarrow \left[a^* \frac{\omega_k}{\chi_k} - \frac{1}{b_k}\right]_0^{s_k^{\text{mask}}}, \quad \forall k \in \mathcal{K}$

Algorithm 3.4 Rate Adaptive Recursive Water-Filling Algorithm

$i \leftarrow 1, W_0 \leftarrow 0, P_1^* \leftarrow 0$
$A_k^{\min} \leftarrow \frac{1}{d_k}, A_k^{\max} \leftarrow s_k^{\text{mask}} + \frac{1}{d_k}, \quad \forall k \in \mathcal{K}$
$\mathcal{A}^* \leftarrow \text{sort}(\mathcal{A}^{\min} \cup \mathcal{A}^{\max})$
while $(P_i^* \leq P/\Delta f)$ **do**
  $i \leftarrow i + 1$
  if $i = 2K + 1$ **then**
    return $s_k \leftarrow s_k^{\text{mask}}, \quad \forall k \in \mathcal{K}$
  end if
  if $A_{i-1}^* \in \mathcal{A}^{\min}$ **then**
    $W_{i-1} = W_{i-2} + 1$
  else
    $W_{i-1} = W_{i-2} - 1$
  end if
  $P_i^* \leftarrow P_{i-1}^* + (A_i^* - A_{i-1}^*)W_{i-1}$
end while
$a^* \leftarrow \frac{P/\Delta f - P_{i-1}^*}{W_{i-1}} + A_{i-1}^*$
return $s_k \leftarrow \left[a^* - \frac{1}{b_k}\right]_0^{s_k^{\text{mask}}}, \quad \forall k \in \mathcal{K}$

no weights and will compare bisection, discrete bit-loading, and GWRF; and the second will look at the generalized case and will compare bisection to GWRF.

3.5.1 Rate Adaptive Water-Filling Algorithm Comparison

In the computational load calculation, $1/d_k$ will be assumed to be a known variable. Moreover, the loop overhead will not be counted. The variable I will be used to denote the number of iterations. Also, the term v_{discrete} will represent the number of bit-loading values each tone can have. As the discrete bit-loading algorithm keeps track of the bit-loading while the bisection and GRWF algorithms do not, each approach will have to generate the final PSD and bit-loading to even out the playing field.

The Recursive Water-Filling approach in Algorithm 3.4 requires K additions at the first step where $\mathcal{A}^{\min}$ and $\mathcal{A}^{\max}$ are produced. The sorting algorithm assumed here is Merge Sort. The number of comparisons needed by the merge sort can be upper bounded by $2K \log_2(2K)$. At each iteration, two comparisons are required, one for the total transmit power constraint and the other for determining from which set the water-filling level

comes from; three additions and subtractions; and one multiplication. Once the water-filling level interval is determined, two additions and one division are needed to obtain the water-filling level. Another K additions and $2K$ comparisons are needed to calculate the PSD. Calculating the bit-loading requires another K additions, K multiplications, and K logarithms.

The Discrete Bit-Loading approach considered here will initially sort every possible change in PSD values. This allows the algorithm to avoid a K-variable search at every iteration. The sorting algorithm will be used again and the number of comparisons for the sorting process is upper bounded by $Kv_{\text{discrete}} \log_2(Kv_{\text{discrete}})$. Each iteration requires three additions and two comparisons. Note that the rate obtained is not essential and the addition for keeping track of it can be ignored.

The bisection approach does not need any initialization operations. At each bisection step, one addition and one multiplication are needed to calculate the water-filling level, K subtraction and $2K$ comparisons are needed to calculate the PSD, $K-1$ additions are needed to calculate the total transmit power, and one comparison is needed for the bisection range reduction. Once the bisection part done, another K additions, K multiplications, and K logarithms are required to calculate the bit-loading.

The computational load for each approach is summarized in Tables 3.1, 3.2, and 3.3. Table 3.4 gives the interpretation of the number of iterations required by each approach. In the GRWF case, the number of iterations is limited by the number of elements in the set of water-filling levels $\mathcal{A}^*$. For the discrete bit-loading approach using constant bit-loading increments, the rate obtained is a linear function of the number of iterations. For example, using unit increments and a symbol rate f_s of 4000 Hz, 5000 iterations are required to obtain a rate of 20Mbit/s. Finally, the water-filling level precision in the bisection approach is a function of the number of iterations. As for the value of v_{discrete}, it depends on the size of the set of allowable bit-loading values. Given a maximum of 16 b/s/Hz and a unit-bit-loading increment, v_{discrete} is equal to 16.

Comparing the computational load, it is clear that the discrete bit-loading approach has the advantage of not needing to calculate the bit-loading values from the resulting PSD thus removing the need to perform the logarithms. However, taking the number of iterations and the value of v_{discrete} into account, the GRWF approach has the advantage of potentially requiring the least number of arithmetic operations and comparisons.

Table 3.1 Number of operations needed by the rate adaptive GRWF approach to obtain the optimal PSD and bit-loading.

	GRWF
$+/-$	$3K + 3I_{\text{GRWF}} + 2$
$\times$	$I_{\text{GRWF}} + K$
$/$	1
$\leq, =, \geq$	$2K \log_2(2K) + 2I_{\text{GRWF}} + 2K$
$\log_2$	K

Table 3.2 Number of operations needed by the discrete bit-loading approach to obtain the optimal PSD and bit-loading.

	Discrete Bit-Loading
$+/-$	$Kv_{\text{discrete}} + 3I_{\text{discrete}}$
$\times$	Kv_{discrete}
$/$	0
$\leq, =, \geq$	$2I_{\text{discrete}} + Kv_{\text{discrete}} \log_2 (Kv_{\text{discrete}})$
$\log_2$	0

Table 3.3 Number of operations needed by the bisection approach to obtain the optimal PSD and bit-loading.

	Bisection
$+/-$	$(2K)I_{\text{bisection}} + K$
$\times$	$I_{\text{bisection}}$
$/$	0
$\leq, =, \geq$	$(1 + 2K)I_{\text{bisection}} + K$
$\log_2$	K

Table 3.4 Interpretation of the number of iterations in Tables 3.1, 3.2, and 3.3.

GRWF	$1 \leq I_{\text{GRWF}} \leq 2K$
Discrete Bit-Loading	$I_{\text{discrete}} = R^{\text{obtained}}/f_s B$
Bisection	$a \pm (\max_{k\in\mathcal{K}}(\frac{1}{d_k} + s_k^{\text{mask}}) - \min_{k\in\mathcal{K}}(\frac{1}{d_k}))/2^{I_{\text{bisection}}}$

Table 3.5 Number of operations needed by each Water-Filling approach to obtain the optimal PSD and bit-loading.

	GRWF	Bisection
$+/-$	$3K + 5I_{\text{GRWF}} + 3$	$(4K-1)I_{\text{bisection}}$
$\times$	$3K + 2I_{\text{GRWF}}$	$(1+3K)I_{\text{bisection}}$
$/$	2	0
$\leq, =, \geq$	$2K\log_2(2K) + 3I_{\text{GRWF}} + 2K$	$(2+2K)I_{\text{bisection}} + K$
$\log_2$	$K + I_{\text{GRWF}}$	$KI_{\text{bisection}}$
2^x	1	0

3.5.2 Generalized Water-Filling Algorithm Comparison

It is in the generalized water-filling case with rate constraints that GRWF shines best relative to bisection. Under this case, the bisection method is heavily penalized with the overhead added by the rate constraint while the GRWF approach receives much smaller increases to its computational load. With the bisection approach, K logarithms are required at each iteration to compare the rate obtained with the target rate. With the GRWF approach, only one new logarithm is required at each iteration. This means that at most $2K$ logarithms are required to obtain a final PSD and another K for the corresponding bit-loading values. The computational load needed to obtain both the PSD s_k and the bit-loading b_k for the bisection and the GRWF approaches are summarized in Table 3.5. The results in Table 3.5 assume that ω_k/χ_k and $1/d_k$ are known, and that the loop overhead is not counted. The number of iterations I are the same as in the RA case and are provided in Table 3.4. The biggest saving for the GRWF approach comes from the reduction in the number of required logarithms by being able to calculate the rate obtained recursively.

3.6 Application of GWRF to Water-Filling-Like Algorithms

The water-filling algorithm can be applied to other algorithms besides IWF. This section presents other instances where the recursive water-filling approach may or may not be used.

Up to now, this chapter applied recursiveness to the water-filling problem in order to improve its performance. The locally-optimal algorithms presented in Chapter 2, DSB and SCALE, have a water-filling-like spectrum update formula that can be generalized in the following form:

$$s_k = \frac{\omega_k}{\lambda + \beta_k} - \kappa_k.$$

One might wonder if the recursive approach can be applied to the above water-filling-like update formula, and more particularly to DSB and SCALE specifically. The answer is no.

The recursive approach cannot be efficiently applied to the water-filling-like algorithms mainly because of the offset β_k on the Lagrange dual variable λ. In GRWF, the water-filling level is found by solving a linear equation. With water-filling-like algorithms, getting λ requires solving a Kth-order polynomial. For the same reasons, the projection approach also fails here.

On the other hand, the discrete bit-loading approach can be applied to water-filling-like algorithms. However, the interpretation of incrementing the frequency tone that can increase its bit-loading with the least power increase does not hold here. This is due to the offset β_k making a non-flat water-fill level. As such, the next frequency tone that gets to increase its bit-loading may not necessarily be the most efficient choice in term of power usage. Thus, instead of looking for the minimal change in PSD, the approach must look at the Lagrange dual variable itself. The discrete bit-loading approach can be interpreted as starting off with an infinitely large Lagrange dual variable and gradually shrinking it until either the total transmit power constraint is met, or until the Lagrange dual variable becomes 0. The discrete bit-loading approach applied to water-filling-like is summarized in Algorithm 3.5. Although Algorithm 3.5 is for water-filling-like algorithms, it still applies to the regular water-filling case and actually generalizes Algorithm 3.1. The change in PSD $\Delta s_k(b_k)$ are the same as ones used in Algorithm 3.1. The variable λ_k is the value of the Lagrange dual variable required to obtain a bit-loading of b_k on frequency tone k and it is defined in (3.12).

$$\lambda_k(b_k) = \begin{cases} \frac{\omega_k d_k}{2^{b_k}} - \beta_k & \text{for DSB} \\ \frac{\omega_k d_k}{2^{b_k}-1} - \beta_k & \text{for SCALE} \\ \frac{\omega_k d_k}{2^{b_k}+d_k\kappa_k-1} - \beta_k & \text{for other water-filling-like algorithms} \end{cases} \tag{3.12}$$

Algorithm 3.5 Discrete Bit-Loading for Water-Filling-Like Algorithms with Constant Bit-Loading Increments

$\mathcal{K} \leftarrow \{1, 2, \ldots, K\}$
$b_k \leftarrow 0 \quad \forall k \in \mathcal{K}$
$s_k \leftarrow 0 \quad \forall k \in \mathcal{K}$
$P^{\text{used}} \leftarrow 0$
$R^{\text{obtained}} \leftarrow 0$
loop
  $k \leftarrow \underset{k \in \mathcal{K}}{\text{argmax}}\, \lambda_k(b_k + B)$
  if $\lambda < 0$ **then**
    break
  else if $P^{\text{used}} + \Delta f\, \Delta s_k(b_k) \geq P$ **then**
    break
  else if $s_k + \Delta s_k(b_k) \geq s_k^{\text{mask}}$ **then**
    $\mathcal{K} \leftarrow \mathcal{K} \setminus k$
  else
    $s_k \leftarrow s_k + \Delta s_k(b_k)$
    $b_k \leftarrow b_k + B$
    $P^{\text{used}} \leftarrow P^{\text{used}} + \Delta f\, \Delta s_k(b_k)$
    $R^{\text{obtained}} \leftarrow R^{\text{obtained}} + f_s$
  end if
end loop

3.7 Conclusion

This chapter overviewed a discrete bit-loading and a projection approach for water-filling with each method having its own advantage over the traditional bisection given certain conditions. Then, the generalized water-filling problem was presented and its recursive formulation derived in order to introduce the GRWF algorithm. The GRWF can solve the water-filling problem within a fixed number of iterations and can use both maximum total transmit power and minimum target rate constraints concurrently with little added computational complexity relative to just using the minimum target rate constraint. Finally, this chapter showed that a similar recursive approach cannot be applied to water-filling like algorithms such as DSB and SCALE. However, to improve over the bisection method, a bit-loading approach for water-filling-like algorithms similar to the one for regular water-filling is introduced.

Chapter 4

Autonomous Spectrum Balancing Using Multiple Reference Lines (ASB-MRL)[1]

Chapter 3 introduced new water-filling methods that improved the Iterative Water-Filling (IWF) algorithm's performance through its complexity. With IWF, each user distributes its allowable transmit power over the available frequency tones such that it is optimal to itself. In some cases, this self-optimization can lead to good results. However, in other cases, IWF can perform poorly and an optimization process that takes every user in the network into account is needed. This network-level optimization can control all the users and it also requires full network information but at the cost of needing to perform the algorithm centrally at a Spectrum Management Center (SMC) or semi-centrally with the use of message-passing system between every user. Examples of centralized algorithms include Optimal Spectrum Balancing (OSB) [22] and Iterative Spectrum Balancing (ISB) [26], and examples of semi-centralized includes Distributed Spectrum Balancing (DSB) [20] and Successive Convex Approximation for Low-complExity (SCALE) [33]. Of the four algorithms, only OSB can guarantee globally-optimal results through exhaustive search whereas ISB gives near-optimal results, and DSB and SCALE produce locally-optimal results. However, as previously stated, these network-level algorithms cannot be fully distributed because of their global control requirements. Moreover, their higher performance comes at the cost of

[1]Part of Chapter 4 has been presented in [41].

higher complexity.

This chapter introduces a compromise between the user-optimal IWF and the network-optimal algorithms by introducing a new algorithm that extends IWF by modifying the underlying water-filling algorithm. Autonomous Spectrum Balancing using Multiple Reference Line (ASB-MRL) algorithms are then introduced as algorithms capable of exploiting that extension hence allowing them to produce results similar to the ones produced by network-level algorithms while keeping a similar complexity to IWF. Unlike the network-level algorithms, ASB-MRL only requires a centralized initialization phase and can then be fully distributed to every user. The performance increase seen in ASB-MRL is obtained by feeding it network information using multiple reference lines. Moreover, the ASB-MRL algorithms can outperform the single-reference-line ASB due to the multiple reference lines containing more information about the network than the single one. However, the choice of reference lines is crucial because they must be able to represent the actual network. As such, this chapter develops three conditions on the set of reference lines for near-optimal performances. Finally, this chapter shows through a Monte Carlo style simulation that the ASB-MRL algorithms can indeed obtain near-optimal performance over randomly generated DSL networks.

In Section 4.1, a comparison between IWF and the optimal algorithm DSB is made to assess how they are different in terms of effective water-filling levels. Section 4.2 uses the differences to introduce an extension to IWF by adding an offset to the Lagrange multiplier in WF in order to give it the potential to produce near-optimal solutions. The ASB-MRL algorithms, ASB-DSB and ASB-SCALE, are introduced in Section 4.3 as a method of generating the Lagrange offsets. Within the same section, the constant offset ASB-MRL algorithms are also presented. Section 4.4 shows the faster convergence speed that ASB-MRL algorithms have over network-level algorithms. The conditions for selecting the reference lines used by ASB-MRL are introduced in Section 4.5. Finally, a Monte Carlo style simulation over an all-CO network and over a mixed CO-RT network is performed in order to show ASB-MRL's capability in achieving near-optimal results.

4.1 IWF and Optimal Algorithms

The IWF algorithm is a simple algorithm because each user only needs to calculate its water-filling level using the interference it measures and its direct channel gain. On the

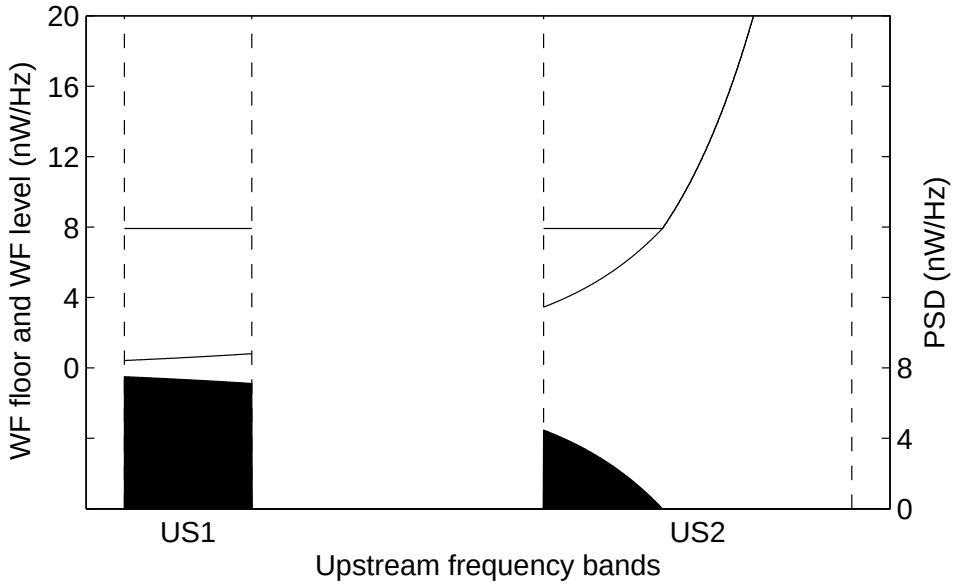
other hand, network-level algorithms require information from all users in order to generate a solution. In order to gain a better understanding on why the network-level algorithms can perform better than IWF, this section sets out three examples that highlights their differences from a water-filling level perspective that will set the basis in the IWF extension. In all three example, DSB is used as the optimal network-level algorithm.

4.1.1 Example 1: Encouraging Frequency Tones with Low Direct Channel Gains

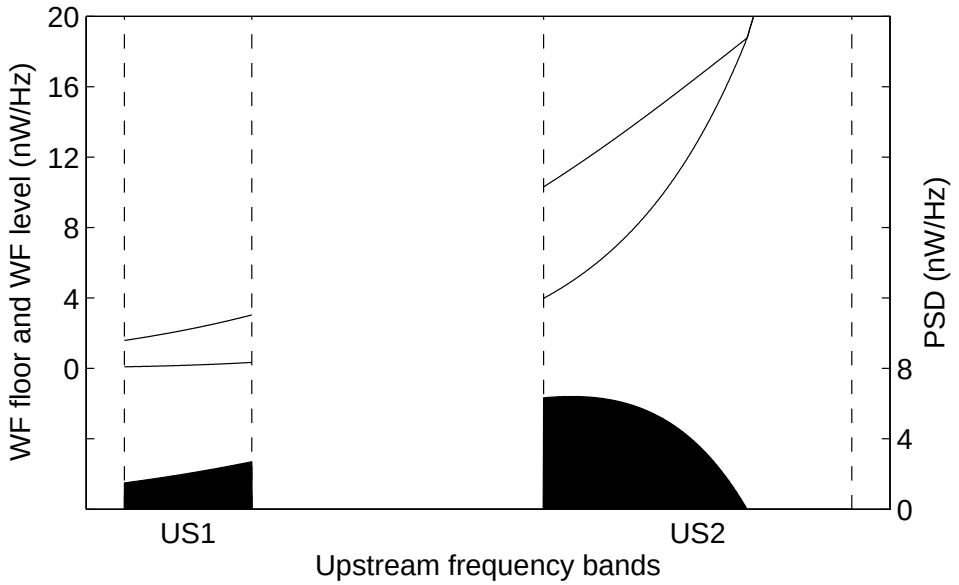
This first example looks at the case where the performance gap between IWF and the locally-optimal algorithm DSB is non-negligible: a network containing 10 identical users with common start and end points, and with line lengths of 900m, transmitting in the upstream bands. The Power Spectral Density (PSD) produced by each algorithm is shown in Figure 4.1. It is quite obvious that DSB allocates more power towards the inferior and higher frequency tones where the crosstalk channel gains are greater and the direct channel gains are lower. This results in DSB using more frequency tones than IWF. However, even though the PSD is not optimal in the user-centric-sense, this push towards the inferior frequency tones grants DSB a sum rate over all users of 64 Mbps which is a 10% gain over IWF's 58 Mbps.

To show the differences in IWF terms, the effective water-filling level and the associated water-filling floor are also provided in Figure 4.1. The effective water-filling level produced by DSB effectively shows that the user-optimal flat level is non-network-optimal. Instead, a slanted level with its higher end over the inferior frequency tones leads towards optimality. Thus, the push towards inferior frequency tones is a characteristic that the extension on IWF must have.

Note that it is worth noting that it appears to be possible to join the PSDs, the effective water-filling levels and bottoms by a smooth and continuous line through the no-transmit gap between the first upstream band, US1, and the second one, US2. This is no coincidence because the channel model these examples are based on produces smooth and continuous channel gains.



(a) IWF



(b) DSB

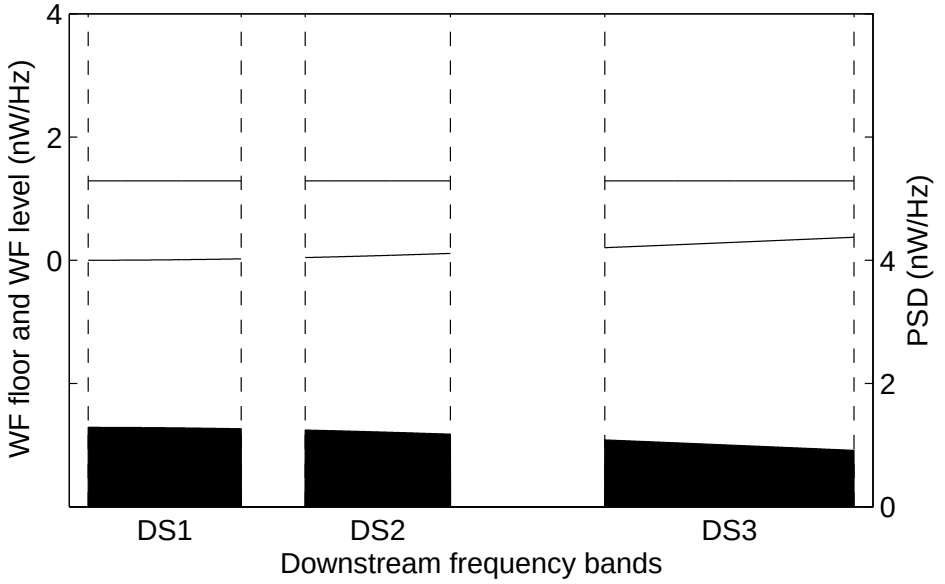
Fig. 4.1 PSD produced by IWF and DSB, and associated effective water-filling level for each user in a 10-user-homogeneous DSL network with line length of 900m in the upstream direction. (Notes: the filled PSD is shown in black, the middle line is the measured interference over direct channel gain ratio, and the top line the effective water-filling level.)

4.1.2 Example 2: Frequency-Selective Water-Filling Level

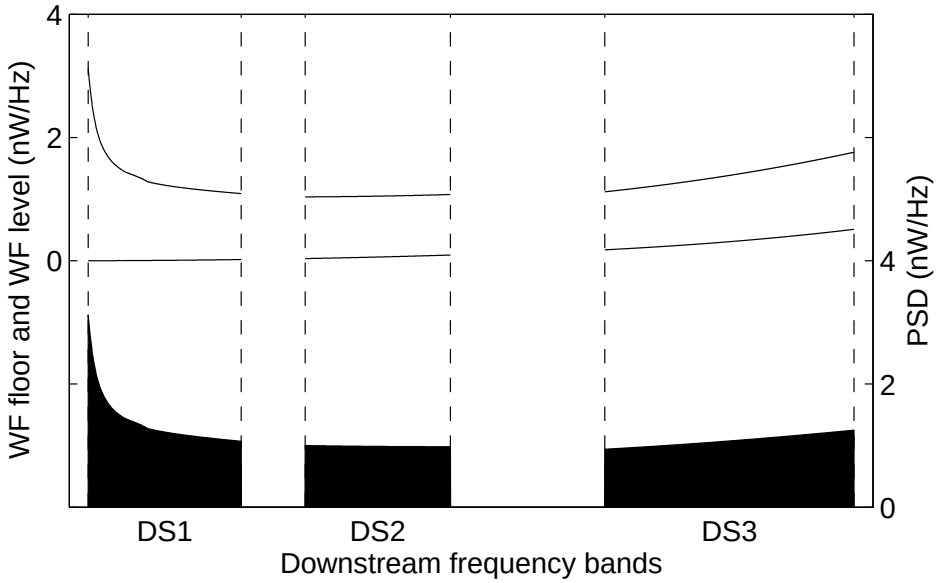
This second example shows that even in cases where the performance gap between IWF and DSB is small, the differences are still noticeable. This example uses a similar network as the one used in the first example but with line lengths of 300m and in the downstream direction. This shorter distance is used because both algorithms will both produce solutions capable of a sum rate of 501 Mbps. Yet, even with identical sum rates, the PSDs shown in Figure 4.2 show that DSB is distributing more power towards the very low frequency tones in DS1 where the direct channel gain is high and crosstalk channel gains are low (superior frequency tones), and also towards the inferior and higher frequency tones. Hence, DSB is taking advantage of the superior channels while still attempting to promote the inferior ones but with no noticeable performance gains. Moreover, the effective water-filling level from this example shown in Figure 4.2 and the one from the first example show that using a frequency-selective level has the potential to improve the performance. Thus, the IWF extension should use a frequency-selective water-filling level.

4.1.3 Example 3: Self Total Transmit Power Restraint

This third example shows another difference between IWF and DSB: self-imposed total transmit power reduction. This total transmit power reduction can be easily shown using a network with a near-far effect where near users can produce significant crosstalk towards the far users. This example uses a network in the upstream direction composed of 10 users with common end points and with line lengths uniformly distributed between 450 and 900 m. In this kind of near-far network, the crosstalk channel gain from the 450-m user to 900-m user is much greater than that on the opposite direction. This is because the 900-m user's signal has to attenuate over 450m before seeing its line has the chance of generating crosstalk onto the 450-m user's line. By maximizing the overall throughput of this network, IWF can only reach a sum rate of 95 Mbps as opposed to DSB's 119 Mbps. As in the the first and second examples, the effective water-filling level produced by DSB presents the same frequency-selective characteristics with slanted level increasing with frequency. What is more remarkable is total transmit power required by each user. The values are shown in Table 4.1. With IWF, all users are transmitting using their full allowable total transmit power of 11.5 dBm. On the other hand, with DSB, the four users with the shortest line lengths are transmitting below their allowable total transmit power, as much as 14 dB less



(a) IWF



(b) DSB

Fig. 4.2 PSD produced by IWF and DSB, and associated effective water-filling level for each user in a 10-user-homogeneous DSL network with line length of 300m in the downstream direction. (Notes: the filled PSD is shown in black, the middle line is the measured interference over direct channel gain ratio, and the top line the effective water-filling level.)

Table 4.1 Total transmit power used by each user in example 3.

	Total transmit power used	
Line length (m)	IWF (dBm)	DSB (dBm)
450	11.5	-3.4
500	11.5	0.4
550	11.5	4.3
600	11.5	8.5
650	11.5	11.5
700	11.5	11.5
750	11.5	11.5
800	11.5	11.5
850	11.5	11.5
900	11.5	11.5

in the 450-m user's case. This shows that reducing the transmit power on certain users will benefit the entire network. Hence, controlling the total transmit power also plays a role in addition to the frequency-selective water-filling level in increasing the performance.

4.1.4 Example Wrap-Up

In the previous three examples, the effective water-filling level associated to optimal algorithms were not flat and followed a general pattern. This effective water-filling level pattern has also been observed in all other simulations regardless of network composition. Hence, using a frequency-selective water-filling level presents a major potential. However, playing with the water-filling level alone is insufficient. A total transmit power reduction process is also required particularly when a near-far effect is present in order to prevent a user with a large crosstalk generation potential from overpowering the other users.

4.2 Extending Water-Filling

Previously, other algorithms have been created using IWF as a backbone while trying to produce optimal PSD patterns. The Selective Iterative Water-filling algorithm (SIW) [38] iteratively applies IWF over a reducing set of frequency tones. This iterative process allows some users to transmit at a higher power over some frequency tones thus leading to an effective water-filling made up of multiple flat levels. However, this method requires message passing between users and IWF has to be performed many times until the results

converge. Multi-Level Water-Filling (MLWF) [42] is another algorithm that tries to produce optimal PSD patterns. With MLWF, the frequency tones are separated into multiple groups each with its own water-filling level. MLWF distributes the power between the sections using a priority scheme in order to replicate the optimal PSD pattern. The drawback of MLWF is that the optimal frequency tone groups are unknown and must be found through an iterative searching algorithm and water-filling is performed at every search iteration. Both MLWF and SIW can only approximate the optimal PSD patterns using multiple flat water-filling lines hence effectively quantizing effective water-filling level. In the MLWF's case, increasing the number of groups in order to better approximate the pattern will only increase the complexity of the algorithm and the number of iterations for convergence.

This section presents an extension on the water-filling algorithm that will allow IWF to produce results similar to the ones produced by the network-level algorithms. The extension achieves this by both modifying the effective water-filling level and introducing a self-imposed total transmit power constraint.

The water-filling algorithm is quite simple in that it only needs to find that water-filling level used in the spectral update formula rewritten here:

$$s_k^n = \left[a^n - \frac{\Gamma \mathrm{int}_k^n}{H_k^{nn}} \right]_0^{s_k^{n,\mathrm{mask}}}$$

where a^n represents the water-filling level and is the inverse of the normalized Lagrange variable λ^n, and $[\cdot]_0^{s_k^{n,\mathrm{mask}}}$ bounds the PSD between 0 and the PSD mask. At first glance, there are two obvious possible methods for creating a frequency-selective level:

1. adding a frequency-selective offset to the water-filling level a^n to obtain $s_k^n = \left[a^n + \mathrm{offset}_k^n - \frac{\Gamma \mathrm{int}_k^n}{H_k^{nn}} \right]_0^{s_k^{n,\mathrm{mask}}}$,
2. or adding a frequency-selective weight to the water-filling level a^n to obtain $s_k^n = \left[w_k^n a^n - \frac{\Gamma \mathrm{int}_k^n}{H_k^{nn}} \right]_0^{s_k^{n,\mathrm{mask}}}$.

With either method, the effective water-filling level can be modified into any shape desired. As for implementing the self-constraint on the total transmit power, users will large outgoing crosstalk channel gains can perform the water-filling procedure such that they meet a target rate instead of using their full allowable total transmit power. However, there still remains a challenge. The offsets and weights are actually dependent on the total transmit

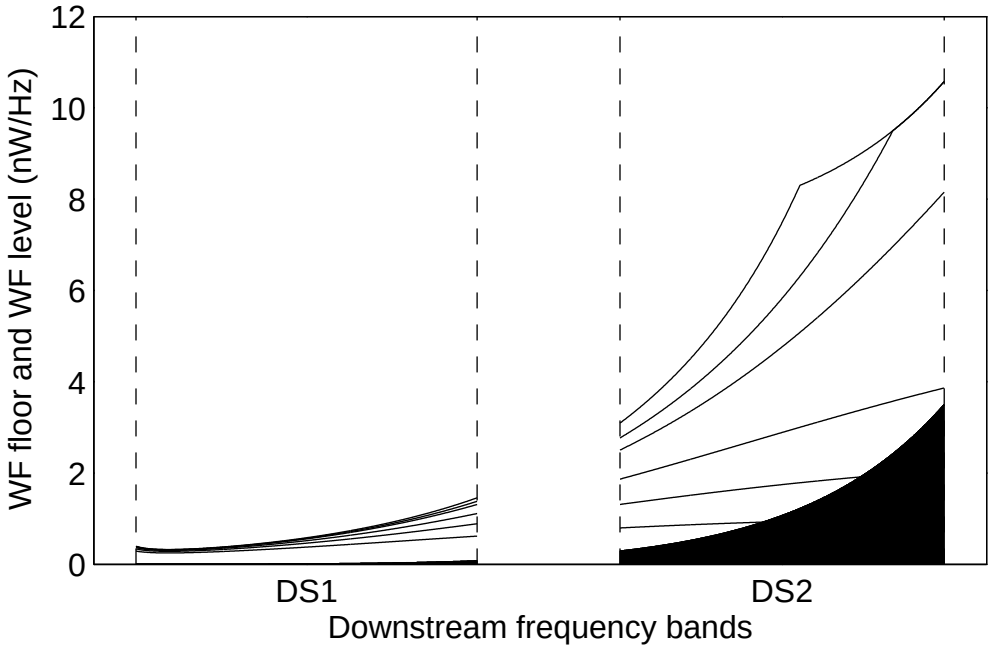


Fig. 4.3 Effective water-filling level for different total transmit power constraints.

power used. Figure 4.3 shows this dependence where the effective water-filling level for various total transmit power values from DSB are displayed. The corresponding network has 10 users with line lengths of 900m and with common start and end points. In the first downstream band (DS1), the effective water-filling level does not vary much as the total transmit power changes. In the second downstream band (DS2), the level changes shape as the total transmit power increases. This shape change is impossible if using a constant offset. Similarly, constant weights will also not work. Thus, with either modification, the user will have to dynamically change the offsets or weights whenever the total transmit power changes which can become rather problematic with fixed-margin users as their total transmit power changes in order to reach a target rate.

A better modification is to add a frequency-selective offset $\Delta\lambda_k^n$ to the Lagrange multiplier as follows:

$$s_k^n = \left[\frac{1}{\Delta\lambda_k^n + \lambda^n} - \frac{\Gamma \mathrm{int}_k^n}{H_k^{nn}}\right]_0^{s_k^{n,\mathrm{mask}}} . \tag{4.1}$$

This Lagrange offset has two benefits. The first benefit is that it acts as a self-imposed PSD mask. When given a sufficiently large total transmit power constraint, the Lagrange

variable λ^n will be set to 0 and the following upper bound on the PSD will be obtained:

$$s_k^n = \min\left(\frac{1}{\Delta\lambda_k^n} - \frac{\Gamma\text{int}_k^n}{H_k^{nn}}, s_k^{n,\text{mask}}\right). \tag{4.2}$$

Thus, by carefully setting the Lagrange offset, we can prevent a user from over-loading specific frequency tones. Also, this upper bound is effectively a self-imposed total transmit power constraint. The second benefit from the Lagrange offset is that it will produce an effective water-filling level that changes dynamically with the total transmit power used. At sufficiently low total transmit power constraints, the resulting Lagrange variable λ^n will be much larger than the Lagrange offset $\Delta\lambda_k^n$ and the modified algorithm will revert to water-filling as shown in the following equation:

$$s_k^n = \frac{1}{\Delta\lambda_k^n + \lambda^n} - \frac{\Gamma\text{int}_k^n}{H_k^{nn}} \xrightarrow{\lambda^n \gg \Delta\lambda_k^n, \forall k \in \mathcal{K}} s_k^n = \frac{1}{\lambda^n} - \frac{\Gamma\text{int}_k^n}{H_k^{nn}}.$$

When using the Lagrange offsets, the effective water-filling level becomes flat at low total transmit power and approaches the upper bounded PSD defined in 4.2 as the total transmit power increases. Thus, the effect of the Lagrange offset is to delay the increase of the effective water-filling over a frequency tone.

In the cases where a user has a target rate to meet but with a self-imposing PSD mask that does not allow such target rate, a weight can be added to the modified water-filling to give

$$s_k^n = \min\left(\frac{w^n}{\Delta\lambda_k^n + \lambda^n} - \frac{\Gamma\text{int}_k^n}{H_k^{nn}}, s_k^{n,\text{mask}}\right).$$

The weight has to be increased until the desired target rate is obtained much like in the DSB FM process discussed in Chapter 2.

There can be many possible ways of generating the Lagrange offset values but this chapter focuses on the use of multiple reference lines as it is based on optimization and it can be derived. Other methods include using supervised machine learning processes such as using decision trees and neural networks to exploit the correlation between the measured interference, channel characteristics, and resulting Lagrange offsets.

4.3 ASB-MRL Algorithms

The previous section showed that adding an offset to the Lagrange variable in the water-filling formulation has the potential of allowing IWF to perform as well as network-level algorithms. In fact, with the Lagrange offset, the modified water-filling approach is similar to the DSB approach where the offset containing information about every other user is generated by a spectrum management center and is then relayed to each user. However, as mentioned previously about semi-centralized algorithms, this offset relaying has two major hurdles: the need and use of a message passing system before every PSD update iteration, and the knowledge of the crosstalk channel gains between every user on every frequency tone. In order to remove these two hurdles, this section introduces the Autonomous Spectrum Balancing using Multiple Reference Lines algorithms (ASB-MRL) which provides a different way of generating the aforementioned offsets. Once derived, the ASB-MRL algorithms have the same form as (4.1) and their Lagrange offsets are a function of the multiple reference lines.

4.3.1 Problem Formulation

The use of multiple reference lines requires the introduction of new variables: let F represent the number of reference lines used, $\mathcal{F} = \{2, 3, \ldots, F+1\}$ be the set of users representing the reference lines, and $n = 1$ be the real user performing the optimization. Note that, like IWF, ASB-MRL is a user-centric algorithm. Thus, the number of reference lines F can vary between users. A constant F is assumed here but the user-dependent-F version of the algorithms can be easily derived. Moreover, each reference line that a user sees can be considered as a virtual user and the combination of the user and the multiple reference lines forms a virtual network. This means that each reference line has the same kind of parameters that an actual user would have. Plus, each reference line has a transmit PSD, and values for the direct and crosstalk channel gains.

This virtual network is effectively the same as an actual network, except that the reference lines have static variables. Therefore, the objective function that each user wants to optimize will be similar to the one used by the network-level optimization algorithms but

with the optimization working only on the real user's variables as follows:

$$\begin{aligned} \max_{s_k^1 \forall k \in \mathcal{K}} \quad & \omega_1 R^1 + \sum_{f \in \mathcal{F}} \omega_f R^f \\ \text{such that} \quad & \Delta f \sum_{k \in \mathcal{K}} s_k^1 \leq P^1 \\ & 0 \leq s_k^1 \leq s_k^{\text{mask},1} \end{aligned} \tag{4.3}$$

where R^f represents reference line f's achievable data rate and where user 1 represents the actual user performing the optimization. However, the formula for $R^f \ \forall f \in \mathcal{F}$ differs from R^1. Since user 1 is real, the actual measured interference int_k^1 over each frequency tone k can be used to calculate the sum rate

$$R^1 = f_s \sum_{k \in \mathcal{K}} \log_2 \left(1 + \frac{1}{\Gamma} \frac{H_k^{11}}{\text{int}_k^1} \right) \tag{4.4}$$

with int_k^1 being the interference measured on frequency tone k. The virtual users $f \in \mathcal{F}$ do not have access to actual measured interference. Instead, the interference on each frequency tone k observed by each reference line $f \in \mathcal{F}$ is is considered as the combined crosstalk contribution from the user and from the virtual network and is calculated a

$$\text{int}_k^f = H_k^{f1} s_k^1 + \sigma_k^f + \sum_{\substack{m \neq f \\ m \in \mathcal{F}}} H_k^{f,m} s_k^m. \tag{4.5}$$

The corresponding sum rate for the reference lines $f \in \mathcal{F}$ is

$$R^f = f_s \sum_{k \in \mathcal{K}} \log_2 \left(1 + \frac{1}{\Gamma} \frac{H_k^{ff}}{\text{int}_k^f} \right). \tag{4.6}$$

It is important to notice that the objective function 4.3 reduces to the same objective function used by IWF in the special case where the number of reference lines F is null. That means that the solution for optimizing 4.3 should give the same result as IWF under the special case. Similarly, in the special case of $F = 1$, the objective function is identical to those of ASB [18] and ASB2 [20], two different versions of the single-reference-line ASB.

4.3.2 ASB-DSB

ASB-DSB is an ASB-MRL algorithm that uses DSB as its base algorithm. This is possible by recognizing that the optimization problem 4.3 is similar to the rate adaptive problem solved by DSB. The only difference lies in each reference line's PSD, $s_k^f \forall f \in \mathcal{F}$, being a constant in the ASB-MRL formulation and an optimization variable in DSB. Nevertheless, this means that there are fewer variables to optimize and that the DSB spectral update formula derived in Chapter 2 can be reused.

Adapting the DSB spectral update formula to the ASB-MRL formulation gives ASB-DSB and is defined as

$$s^1 = \left[\frac{\omega^1}{\lambda^1 + \sum_{f \in \mathcal{F}} \left(\omega^f H_k^{f1} \left(\frac{1}{\text{int}_k^f} - \frac{1}{\text{rec}_k^f} \right) \right)} - \frac{\Gamma \text{int}_k^1}{H_k^{11}} \right]_0^{s_k^{\text{mask},1}} \tag{4.7}$$

where

$$\text{rec}_k^f = \text{int}_k^f + \frac{H_k^{f1}}{\Gamma} s_k^f. \tag{4.8}$$

As mentioned previously, the special cases of using null and one reference line ($F = 0$ and $F = 1$), ASB-DSB should revert to IWF and ASB, respectively. In the null case, the offset to λ^1 disappears and the update formula becomes identical to the one derived for IWF in Chapter 2. In the case of a single reference line, the update formula becomes equivalent to the update formula derived for ASB2 in [20]. Therefore, ASB-DSB generalizes both IWF and ASB2.

4.3.3 ASB-SCALE

So far, most of this chapter has only compared IWF with DSB. The offset on the Lagrange dual variable is also present in SCALE and all previous analysis still holds. As such, we can create an ASB-MRL algorithm from SCALE in order to solve for the optimization problem 4.3.

In a similar manner to the process from DSB to ASB-DSB, this transformation also uses the similarity between the optimization problem used by ASB-MRL and that used by SCALE. This allows the reuse of SCALE's spectral update formula in order to develop

ASB-SCALE. The ASB-SCALE spectral update formula is

$$s_k^1 = \left[\frac{\omega^1 \alpha_k^1}{\lambda^1 + \sum_{f \in \mathcal{F}} \frac{\omega^f \alpha_k^f H_k^{fn}}{\text{int}_k^f}} \right] \tag{4.9}$$

where

$$\alpha_k^1 = \frac{\frac{H_k^{11} s_k^1}{\Gamma \text{int}_k^1}}{1 + \frac{H_k^{11} s_k^1}{\Gamma \text{int}_k^1}}$$
$$\alpha_k^f = \frac{\frac{H_k^{ff} s_k^f}{\Gamma \text{int}_k^f}}{1 + \frac{H_k^{ff} s_k^f}{\Gamma \text{int}_k^f}}.$$

The difference between α_k^1 and $\alpha_k^f \forall f \in \mathcal{F}$ lies in the source of the observed interference values where the real user uses measured values and the reference lines calculating theirs.

As in SCALE, the values of α_k^1 and α_k^f do not need to be updated at every iteration. They should also be initialized to 1 as to represent an infinite signal to interference plus noise ratio. Similarly, s_k^f should be initialized to zero. On the second iteration, the reference line PSDs s_k^f can take their constant values.

In the special case of zero reference lines, ASB-SCALE does in fact become IWF but under a different form. Instead of obtaining the IWF spectral update formula derived in Chapter 2, ASB-SCALE transforms into the RA-WF algorithm derived in [33].

4.3.4 Good Virtual Network Approximation

Up to this point, each user using the ASB-DSB and ASB-SCALE algorithms needs to calculate K F-dimension linear equations at every iteration for updating their reference line's observed interference value, int_k^f. This leads to a need of constantly updating the Lagrange variable λ^1's offset.

A simplification can be made to the ASB-MRL algorithms by assuming that the multiple reference lines already represent the overall network well. Under this assumption, we can neglect the user's contribution to the reference line's interference calculation because the crosstalk contribution from the other reference lines already represent the crosstalk composition. That is, the $H_k^{f,1} s_k^1$ term can be omitted when calculating int_k^f. Under this

Algorithm 4.1 Constant Offset ASB-DSB Algorithm

Get reference line parameters: $s_k^f, H_k^{f,1}, H_k^{1,f}, H_k^{f,f}, \quad \forall f \in \mathcal{F}$
$\text{int}_k^f \leftarrow \sigma_k^f + \sum_{\substack{m \neq f \\ m \in \mathcal{F}}} H_k^{f,m} s_k^m, \quad \forall f \in \mathcal{F}$
$\text{rec}_k^f = \text{int}_k^f + \frac{H_k^{f,1}}{\Gamma} s_k^f, \quad \forall f \in \mathcal{F}$
$\Delta\lambda_k \leftarrow \sum_{f \in \mathcal{F}} (\omega^f H_k^{f,1} (\frac{1}{\text{int}_k^f} - \frac{1}{\text{rec}_k^f}))$
repeat
  Measure interference int_k^1
  Find $\lambda \in [0, \infty]$ such that $\sum_{k \in \mathcal{K}} \left[\frac{\omega^1}{\lambda + \Delta\lambda_k} - \frac{\Gamma \text{int}_k^1}{H_k^{1,1}} \right]_0^{s_k^{\text{mask},1}} = P^1$
  if unfeasible **then**
    $\lambda \leftarrow 0$
  end if
  $s_k^1 \leftarrow [\frac{\omega^1}{\lambda + \Delta\lambda_k} - \frac{\Gamma \text{int}_k^1}{H_k^{1,1}}]_0^{s_k^{\text{mask},1}}$
until convergence

assumption, the Lagrange variable's offset becomes constant and reduces the overall complexity per iteration of ASB-DSB and ASB-SCALE to that of IWF. Hence, the simplified version of ASB-DSB and ASB-SCALE use

$$\text{int}_k^f = \sigma_k^f + \sum_{\substack{m \neq f \\ m \in \mathcal{F}}} H_k^{fhem} s_k^m \tag{4.10}$$

for calculating the reference line's measured interference. This leads to the constant offset ASB-DSB in Algorithm 4.1 and the constant offset ASB-SCALE in Algorithm 4.2.

4.4 ASB-MRL with Full Network Topology Information

The previous section provided the ASB-DSB and ASB-SCALE spectral update formula for using ASB-MRL. Yet, the parameters associated to the reference lines still need to be known. More specifically, the PSD used by each reference, $s_k^f \forall k \in \mathcal{K}$ and $f \in \mathcal{F}$, and the crosstalk channel gains between each reference line, $H_k^{mf} \ \forall k \in \mathcal{K}$ and $m, f \in \mathcal{F}$, and between the reference lines and the real user, H_k^{1f} and H_k^{f1}, are required. Yet, before venturing towards the choice of reference lines, this section explores the potential of ASB-MRL algorithms in the simple case where the full network topology is known to each user.

Algorithm 4.2 Constant Offset ASB-SCALE Algorithm

Get reference line parameters: $s_k^f, H_k^{f,1}, H_k^{1,f}, H_k^{f,f}, \quad \forall f \in \mathcal{F}$

$\text{rec}_k^f = \sigma_k^f + \sum_{\substack{m \neq f \\ m \in \mathcal{F}}} H_k^{f,m} s_k^m + \frac{H_k^{f,1}}{\Gamma} s_k^f, \quad \forall f \in \mathcal{F}$

$\alpha_k^n \leftarrow \frac{\frac{H_k^{f,1}}{\Gamma} s_k^f}{\text{rec}_k^f}, \quad \forall f \in \mathcal{F}$

$\Delta\lambda_k \leftarrow \sum_{f \in \mathcal{F}} \frac{\omega^f \alpha_k^f H_k^{f,1}}{\text{int}_k^f}$

repeat

  Measure int_k^1 and rec_k^1

  $\alpha_k^1 \leftarrow \frac{\text{rec}_k^1 - \text{int}_k^1}{\text{rec}_k^1}$

  if first iteration **then**

    Use $\alpha_k^1 = \alpha_k^f = 1, \forall f \in \mathcal{F}, k \in \mathcal{K}$

    Use $\Delta\lambda_k = \sum_{f \in \mathcal{F}} \frac{\omega^f H_k^{f,1}}{\text{int}_k^f}$

  end if

  Find $\lambda \in [0, \infty]$ such that $\sum_{k \in \mathcal{K}} [\frac{\omega^1 \alpha_k^1}{\lambda + \Delta\lambda_k}]_0^{s_k^{\text{mask},1}} = P^1$

  if unfeasible **then**

    $\lambda \leftarrow 0$

  end if

  $s_k^1 \leftarrow [\frac{\omega^1 \alpha_k^1}{\lambda + \Delta\lambda_k}]_0^{s_k^{\text{mask},1}}$

until convergence

With the full network topology, each user will know the relative position of every user within the network hence forming a possible set of reference lines.

In the simplest case, we assume that the full network topology information is known to each user by giving every other user's line length and relative position. Under this assumption and the slow time-varying property of the channel, channel models can be used to approximate the direct and crosstalk channel gains. Then, depending on how much information is known about the other users, various approaches can be taken for obtaining the PSD used by other users. In the worst case, the PSD is not exchanged between the users. Hence, each user will need to approximate the other user's PSD. One good way to approximate it is to simulate the network by applying DSB or SCALE to a virtual network made from the approximated channel values.

In all cases, using multiple reference lines results in a new variable that is linearly dependent on the user's own PSD. Thus, this effectively trades message passing and full channel knowledge requirement in a locally-optimal solution for a method of approximating

the channel values and the PSD's in a near-optimal solution. Additionally, the overall complexity of algorithm reduces from $\mathcal{O}(KN^2)$ at every iteration with DSB and SCALE to $\mathcal{O}(KN)$ at every iteration using the variable offset, or $\mathcal{O}(K)$ using the constant offset, and $\mathcal{O}(KN^2)$ at initialization for ASB-DSB and ASB-SCALE.

By using a channel model that can approximate the network well and with the full network topology information and user-constraints known, it is very likely that ASB-DSB and ASB-SCALE can reach a similar performance as their locally-optimal counterparts. However, using this method can be redundant given all the information since a near-optimal PSD is already produced when calculating the reference line's PSD. However, the ASB-MRL algorithms should still be used because they also take into account the actual measured interference and direct channel gains.

What is more particular with ASB-DSB and ASB-SCALE is their faster convergence speed relative to their network-level counterparts, DSB and SCALE. Unlike the network-level algorithms, these ASB-MRL algorithms have already initialized their Lagrange variable λ^1's offset with approximate values while DSB and SCALE must iteratively update their Lagrange offsets. To illustrate this faster convergence, the sum rate in the upstream data transmission for a sample network is plotted against the iteration count. The network consists of 11 users connected to same common CO and with line lengths uniformly ranging from 500 to 1000m. Figure 4.4 shows the sum rate evolution for ASB-DSB and DSB while Figure 4.5 shows the sum rate evolution for ASB-SCALE and SCALE. Both ASB-MRL algorithms are able to reach their peak performance in two iterations while DSB and SCALE take over 10 iterations.

4.5 Reference Lines

There are many possible ways of building a set of reference lines. In the previous section, the set of reference lines represented the actual network. However, the full network topology is not always available. Moreover, it can be advantageous to build a set containing fewer reference lines in order to simplify the initialization procedure where the Lagrange offset $\Delta\lambda_k^n$ is calculated. Instead of proposing potentially many different ways to build a set, this section unveils three conditions on the set of reference lines that, when met, will allow the ASB-MRL algorithms to achieve near-optimal performances.

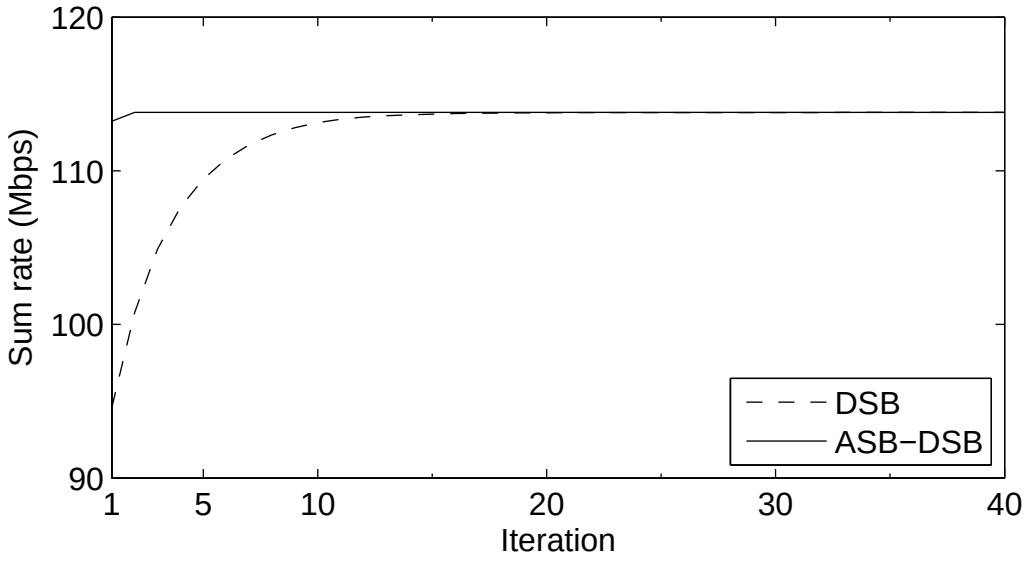


Fig. 4.4 Sum rate versus iteration count for ASB-DSB and DSB.

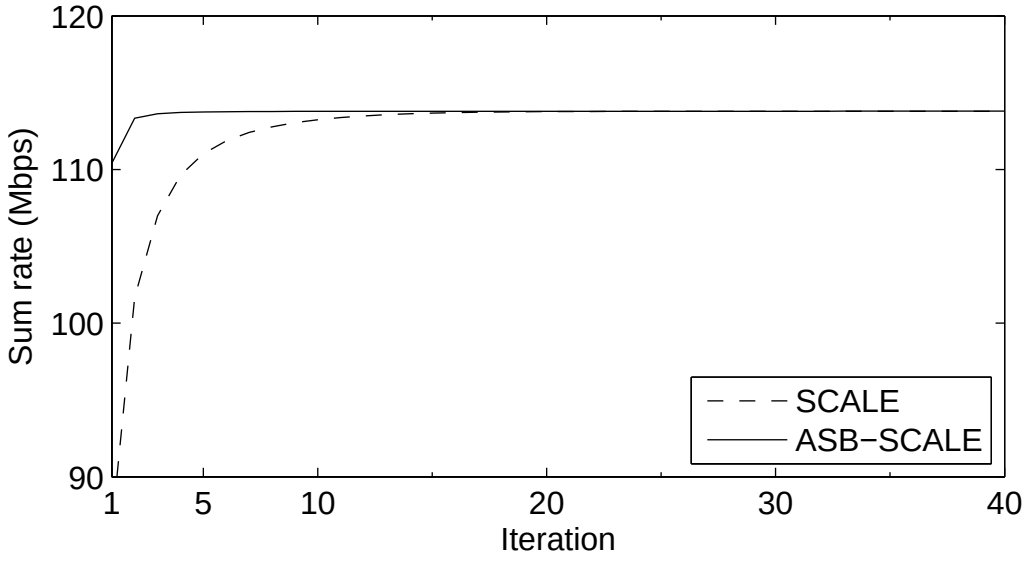


Fig. 4.5 Sum rate versus iteration count for ASB-SCALE and SCALE.

4.5.1 Bit-Loading Coverage Condition

The bit-loading coverage condition exists to try to have a non-zero Lagrange offset $\Delta\lambda_k^n$ on as many frequency tones as possible. On regions with null offsets, the ASB-MRL algorithms will behave like the traditional water-filling. That is, users will be allowed to have a selfish behavior. To prevent null offset values, the reference lines must transmit on as many frequency tones as possible. Ideally, one way to meet this condition is by satisfying

$$\sum_{f\in\mathcal{F}} s_k^f > 0, \quad \forall k \in \mathcal{K}.$$

Often, it is not useful to have a non-zero PSD over every frequency tone. It is quite possible that in many cases some frequency tones have extremely poor channel characteristics for all users. Thus, it is pointless to impose the condition on them as it is very unlikely that a user will ever use them. Similarly, in a far reach network where all users have long line lengths, it is quite likely that the higher frequency tones are unusable by any user and the Lagrange offset value over the higher frequency tone regions will have no impact.

4.5.2 Power Restraint Condition

The set of reference lines should represent the virtual network in order to generate a good Lagrange offset. This representation must not only represent the network through its channel characteristics, but also through the PSD. The reference lines must have a PSD that is typical in the type of network they represent.

In real networks, the power restraint condition comes into play in scenarios where strong and weak users coexist. A strong user is typically one with good direct channel gains, low incoming crosstalk gains, and high outgoing crosstalk gains. On the other hand, a weak user is typically the opposite with poor direct channel gains, high incoming crosstalk gains, and low outgoing crosstalk gains. Thus, if both strong and weak users are allowed to use their total transmit power in its entirety, then the weak user will be receiving high interfering signals relative to its own signal while the strong user receives little interference over its already strong signal. One example of this kind of scenario is the near-far effect discussed in Chapter 2 and pictured in Figure 2.6. Without proper power restraints, the near user in Figure 2.6 will generate crosstalk that will overwhelm the far user's signal power as demonstrated through the size of the arrows.

Like in real networks, the strong-weak coexistence also has an importance in the virtual network since it affects the Lagrange offset. However, the effects of total transmit power reduction on multiple reference lines are different from that in the single-reference-line ASB. Similarly, the effects of power restraint on global optimization using algorithms such as DSB and SCALE are different from user-centric optimization like IWF.

In IWF, each user needs only their direct channel gain and measured interference values on every frequency tone. If another user, which we shall call the interferer, decides to load more power over some bandwidth, then the interference measured by the optimizing user will also increase over the same bandwidth. To adapt to this new interference pattern, the IWF procedure will reduce the power on the frequency tones with higher crosstalk and re-allocate the power to the frequency tones with lower crosstalk. The adaptation process from IWF will happen if the interferer increases its total transmit power over its whole PSD. Thus, from a frequency tone perspective, an increase of power from the interferer will translate in the user avoiding that same frequency tone.

The same avoidance issue comes up in ASB where only one reference line is used. Recall that the spectral update formula using one reference line is

$$s_k^1 = \frac{\omega^n}{\lambda^n + \Delta\lambda_k^n}$$

where

$$\Delta\lambda_k^n = \omega^f H_k^{f1}\left(\frac{1}{\text{int}_k^f} - \frac{1}{\text{int}_k^f + H_k^{ff} s_k^f}\right)$$

and $n = 1$ corresponds to the user performing ASB and $n = f$ the reference line. The Lagrange offset $\Delta\lambda_k^n$ is always non-negative. Its derivative over s_k^f,

$$\frac{\partial \Delta\lambda_k^n}{\partial s_k^f} = \frac{\omega^f H_k^{f1} H_k^{ff}}{(\text{int}_k^f + H_k^{ff} s_k^f)^2},$$

is also non-negative when $s_k^f \geq 0$ since all other values are also non-negative. Thus, increasing the reference line's PSD s_k^f will always result in a Lagrange offset increase. And, as discussed in Section 4.2, increasing the Lagrange offset on a frequency tone will discourage the user from transmitting over that same frequency tone. Moreover, increasing the reference line's PSD over all frequency tones will severely limit the user's achievable rate. Therefore, not using proper total transmit power restraints in ASB will overly restrict

the optimizing user.

In the multiple reference line using ASB-DSB, an increase in the reference line's PSD does not always equate to frequency tone avoidance. Recall that the Lagrange offset in the multiple reference line is given by:

$$\Delta\lambda_k^n = \sum_{f\in\mathcal{F}} \omega^f H_k^{f1}\left(\frac{1}{\text{int}_k^f} - \frac{1}{\text{rec}_k^f}\right).$$

Unlike the single reference line case, increasing the PSD on a given frequency tone k for a given reference line will not always increase the Lagrange offset because the PSD change will impact every other reference line's measured interference. Taking the derivate over the PSD over one of the reference line gives

$$\begin{aligned}\frac{\partial\Delta\lambda_k^n}{\partial s_k^f} &= \frac{\omega^f H_k^{f1}H_k^{ff}}{(\text{rec}_k^f)^2} - \sum_{\substack{m\in\mathcal{F}\\ m\neq f}} \omega^m H_k^{m1}H_k^{mf}\left(\frac{1}{(\text{int}_\text{k}^\text{m})^2} - \frac{1}{(\text{rec}_\text{k}^\text{m})^2}\right)\\ &= \frac{\omega^f H_k^{f1}H_k^{ff}}{(\text{rec}_k^f)^2} - \sum_{\substack{m\in\mathcal{F}\\ m\neq f}} \frac{(2^{2b_k^m}-1)\omega^m H_k^{m1}H_k^{mf}}{(\text{rec}_\text{k}^\text{m})^2}\end{aligned} \tag{4.11}$$

where the term b_k^m is the bit-loading corresponding to the reference line m. The first term is identical to the one obtained in ASB. The second term comes from the change in interference measured by the other reference lines and works against the increase from the first term. This relation shows that increasing one reference line's PSD can actually encourage the optimizing user to transmit more. Hence, increasing the PSD on one reference line can decrease the Lagrange offset thus encouraging the optimizing user to use an even higher transmit PSD and also over-relaxing the user's self power restraint.

A similar analysis can be done using the Lagrange offset on ASB-SCALE. Recall that the Lagrange offset for ASB-SCALE is obtained using

$$\begin{aligned}\Delta\lambda_k^1 &= \sum_{f\in\mathcal{F}} \frac{\omega^f \alpha_k^f H_k^{fn}}{\text{int}_k^f}\\ &= \sum_{f\in\mathcal{F}} \frac{\omega^f H_k^{fn}H_k^{ff}}{\text{int}_k^f \text{rec}_k^f}.\end{aligned}$$

The second step simply rewrites α_k^f in terms of int_k^f and rec_k^f. Taking the derivative over s_k^f gives

$$\frac{\partial \Delta\lambda_k^n}{\partial s_k^f} = \frac{\omega^f H_k^{f1} H_k^{ff}}{(\text{rec}_k^f)^2} - \sum_{\substack{m\in\mathcal{F} \\ m\neq f}} \omega^m H_k^{m1} H_k^{mf} \frac{2\text{int}_k^m + H_k^{mm} s_k^m}{\text{int}_k^m \text{rec}_k^m}. \tag{4.12}$$

Notice the similarities between (4.11) and (4.12). In both cases, a positive change is contributed by the reference line f and a negative change by all other reference lines. Moreover, the negative component is proportional to the crosstalk gain H_k^{mf}, from the reference line that's increasing its PSD towards every other reference line. Thus, it is quite likely that increasing the PSD on reference lines with the potential of generating significant crosstalk towards other reference line will result in encouraging the optimizing user to also increase its PSD. Hence, reference lines with large outgoing crosstalk power gains must use a more restrictive total transmit power constraints.

4.5.3 Virtual Network Self-Representation Condition

The ASB-MRL algorithms require the measured interference measured by each reference line. Under the good virtual network approximation, the contribution to this measured interference comes from the other reference lines only. Thus, the virtual network needs to be representative of the network it approximates from the point-of-view of each reference line.

Figure 4.6 shows an example in which a network contains a mixture of multiple strong and weak users. The virtual network approximates it by using only one strong and one weak reference line. When calculating the measured interference, the strong reference line will only receive crosstalk from the weak one. However, in the actual network, a strong user will receive crosstalk from both strong and weak users. Even if the good virtual network approximation is not used, the virtual network self-representation condition will still not hold. Assuming that the user performing the optimization can be categorized as a weak user, then the strong reference line will receive crosstalk from a weak user and a weak reference line. Thus, the virtual network in the example does not contain a good set of reference lines.

The reference line needing reference-line condition is particularly important when the virtual network contains a small number of reference lines. Virtual networks containing many reference lines that also well represent the actual network will usually meet this

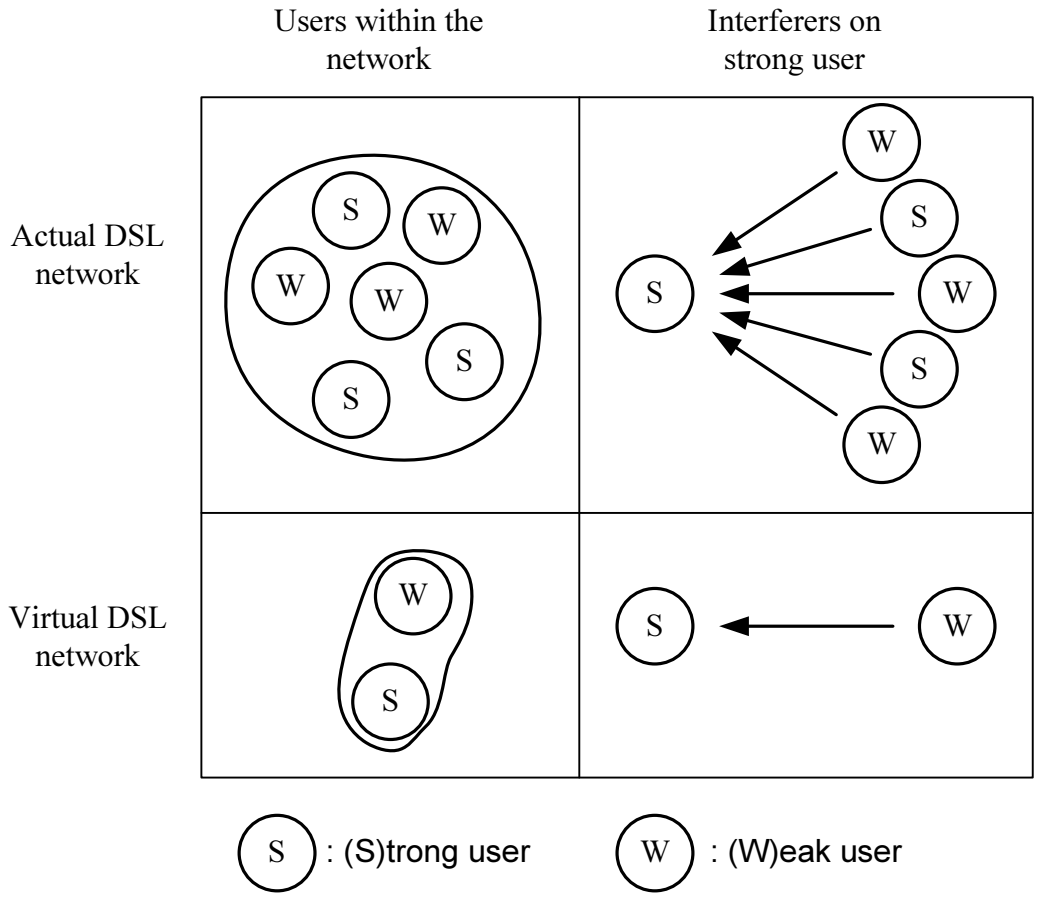


Fig. 4.6 Example of a misrepresentation of the network by the virtual network.

criterion because the measured interference measured by each reference line will have many diversified crosstalk sources. The problem arises in smaller sets where the crosstalk source seen by each reference line is misrepresented. One good solution to solve this problem is to increase the set of reference lines by multiplying it. That is, if the original set contained one long and one short reference line, then doubling the set will give a virtual network with two identical long and two identical short reference lines.

To know whether or not it is necessary to multiply the set of reference line, we must determine if all the reference lines within the set have a well represented interference. The simplest method is to approximate the effect of repeating a single reference line. If repeating

reference line f causes the Lagrange offset to increase by

$$\omega^f H_k^{f1}\left(\frac{1}{\text{int}_k^f}-\frac{1}{\text{rec}_k^f}\right),$$

which corresponds to its original contribution to the Lagrange offset, then repeating has no effect on f's measured interference. Of course, this is highly unlikely without any signal level coordination. Thus, this reference line already has a well-represented measured interference. Note that the original contribution to the Lagrange offset corresponds to the upper bound on the change on the Lagrange offset. The change can also be negative showing that the virtual crosstalk sources to f are not representative of the true crosstalk courses. This can be seen by reformulating the Lagrange offset using the repeating factor $R>1$ for the f's reference line as such:

$$\begin{aligned}\Delta\lambda_k^1(R) = R\omega^f H_k^{f1}&\left(\frac{1}{\text{int}_k^f+(R-1)H_k^{ff,\text{crosstalk}}s_k^f}-\frac{1}{\text{rec}_k^f+(R-1)H_k^{ff,\text{crosstalk}}s_k^f}\right)\\ &+\sum_{\substack{m\in\mathcal{F}\\ m\neq f}}\omega^m H_k^{m1}\left(\frac{1}{\text{int}_k^m+(R-1)H_k^{mf}s_k^f}-\frac{1}{\text{rec}_k^m+(R-1)H_k^{mf}s_k^f}\right).\end{aligned}$$

The $H_k^{ff,\text{crosstalk}}$ represents the crosstalk from f to its twin. Taking the derivative and evaluating it at $R=1$, for the potential change incurred by doubling the reference line f, gives

$$\begin{aligned}\left.\frac{\partial\Delta\lambda_k^1(R)}{\partial R}\right|_{R=1} &= \omega^f H_k^{f1}\left(\frac{1}{\text{int}_k^f}-\frac{1}{\text{rec}_k^f}\right)-\omega^f H_k^{f1}H_k^{ff,\text{crosstalk}}s_k^f\left(\frac{1}{(\text{int}_k^f)^2}-\frac{1}{(\text{rec}_k^f)^2}\right)\\ &\quad-\sum_{\substack{m\in\mathcal{F}\\ m\neq f}}\omega^m H_k^{m1}H_k^{mf}s_k^f\left(\frac{1}{(\text{int}_k^m)^2}-\frac{1}{(\text{rec}_k^m)^2}\right).\end{aligned}$$

The first two terms are of particular interest. They represent the contribution to the derivative from the reference line being multiplied and its multiples while the third term is the negative contribution from all other reference lines. With the first two terms, it can be determined if there is a need to include self-crosstalk on that reference by multiplying it. If the self-crosstalk gain $H_k^{ff,\text{crosstalk}}$ is null, then the second term disappears and the change induced by the first two terms is identical to the reference line's contribution to

the Lagrange offset. Hence, multiplying is not necessary as it will have no effect on the interference measured by reference line f. On the other hand, if the self-crosstalk gain is sufficiently large such that

$$H_k^{ff,\text{crosstalk}} s_k^f \left(\frac{1}{(\text{int}_k^f)^2} - \frac{1}{(\text{rec}_k^f)^2} \right) > \left(\frac{1}{\text{int}_k^f} - \frac{1}{\text{rec}_k^f} \right),$$

then multiplying will have a noticeable effect on reference line f's interference. The larger the gap in the inequality, the greater the effect on interference. Simplifying the condition gives

$$H_k^{ff,\text{crosstalk}} > \frac{\text{int}_k^f \text{rec}_k^f}{(\text{int}_k^f + \text{rec}_k^f) s_k^f}.$$

For cases where the bit-loading b_k^f is high, that is $H_k^{ff} s_k^f \gg \text{int}_k^f$, the condition approximately reduces into

$$H_k^{ff,\text{crosstalk}} > \frac{\text{int}_k^f}{s_k^f}. \tag{4.13}$$

A similar analysis can be performed using the Lagrange offset from ASB-SCALE where including the repeating factor R on reference line f changes the Lagrange offset into

$$\begin{aligned} \Delta\lambda_k^1(R) = & \frac{R\omega^f H_k^{f1} H_k^{ff,\text{direct}}}{(\text{int}_k^f + (R-1) H_k^{ff,\text{crosstalk}} s_k^f)(\text{rec}_k^f + (R-1) H_k^{ff,\text{crosstalk}} s_k^f)} \\ & + \sum_{\substack{m \in \mathcal{F} \\ m \neq f}} \frac{\omega^m H_k^{m1} H_k^{mf}}{(\text{int}_k^m + (R-1) H_k^{mf} s_k^f)(\text{rec}_k^m + (R-1) H_k^{mf} s_k^f)}. \end{aligned}$$

Taking its derivative over R and evaluating it at $R = 1$ gives

$$\begin{aligned} \left. \frac{\partial \Delta\lambda_k^1(R)}{\partial R} \right|_{R=1} = & \frac{\omega^f H_k^{f1} H_k^{ff,\text{direct}}}{\text{int}_k^f \text{rec}_k^f} - \frac{\omega^f H_k^{f1} H_k^{ff,\text{direct}} H_k^{ff,\text{crosstalk}} s_k^f (\text{int}_k^f + \text{rec}_k^f)}{(\text{int}_k^f \text{rec}_k^f)^2} \\ & - \sum_{\substack{m \in \mathcal{F} \\ m \neq f}} \frac{\omega^m H_k^{m1} H_k^{mm} H_k^{mf} s_k^f (\text{int}_k^m + \text{rec}_k^m)}{(\text{int}_k^m \text{rec}_k^m)^2}. \end{aligned}$$

Looking at the contribution from reference line f and its multiples, the condition for re-

peating is

$$\frac{H_k^{ff,\text{crosstalk}} s_k^f (\text{int}_k^f + \text{rec}_k^f)}{\text{int}_k^f \text{rec}_k^f} > 1.$$

Simplifying the condition gives the condition as in the ASB-DSB case:

$$H_k^{ff,\text{crosstalk}} > \frac{\text{int}_k^f \text{rec}_k^f}{(\text{int}_k^f + \text{rec}_k^f) s_k^f}.$$

To determine if repeating is necessary, the repeating condition, (4.13), is used. Satisfying it shows that repeating a reference line does significantly affect the contribution from that line towards the Lagrange offset. If met, then the reference line is repeated. However, the other reference lines must also be repeated in order to preserve the virtual network proportions. After repeating the reference line set, the condition should be tested again to verify if repeating is still necessary.

4.6 Partial Network Topology Simulation Results

This section assesses the performance of the two constant offset ASB-MRL algorithms, ASB-DSB and ASB-SCALE, using virtual networks containing just a few reference lines. Two different types of networks are used for the simulations: an all-CO network where all users connect to the same CO, and a mixed-CO-RT network where some users are connected to a RT and the remaining to the CO. In the all-CO case, the near-far effect is observed in the upstream bandwidths due to some users being closer to the CO than others. Hence, it is expected that ASB-MRL should perform much better than IWF. In the downstream direction, simpler algorithms like IWF should perform close to optimality. Thus, the downstream direction will serve as a test to ensure that ASB-MRL does not under-perform. The mixed-CO-RT has the near-far effect on both the upstream and downstream direction. Furthermore, the network composition is much more varied since the lines originate from two different sources, the CO and the RT. It is in the mixed-CO-RT where optimal algorithms can really outperform the static spectrum management algorithms.

In both simulation scenarios, 100 random network realizations are generated to test the ASB-MRL algorithms against their network-level counterparts, DSB and SCALE. The rate adaptive IWF and ASB are also run in order to make sure that even if the ASB-MRL algorithms perform poorly with a poorer choice of reference line, they can still improve upon the other user-level algorithms.

Table 4.2 DSL simulation parameters.

DMT symbol rate f_s	4 kHz
DMT tone spacing Δf	4.3125 kHz
SNR gap Γ	12.8 dB
Maximum total transmit power P^n	11.5 dBm
Wire diameter (gauge)	0.4 mm (26 AWG)
Noise variance $\sigma_k^n, \forall n \in \mathcal{N}, k \in \mathcal{K}$	-140 dBm

The DSL network parameters presented in Table 4.2 are the ones used for all simulations. The SNR gap Γ is obtained by using a target symbol error probability of 10^{-7} or less, a coding gain of 3dB, and a noise margin of 6dB.

In the results for each algorithm in both simulations, the mean number of iterations required by the algorithms to be within 98% of the converged sum rate will be provided. The 98% is preferred over the 99% and the 100% cases as in those cases, the number of iterations associated to DSB and SCALE are often numbered in the 100s. With DSB and SCALE, such high number of iterations to convergence is impractical since the interference must be measured at every step and may take over a minute to perform. This means that 100s of iterations may practically take hours to perform.

4.6.1 All-CO Network Simulation

The all-CO network has 25 users all connected to the same CO. Each user's line length is taken from a uniform distribution lower bounded at 500m and upper bounded at 1000m. The network topology is pictured in Figure 4.7. In this kind of network, it is expected for IWF to perform relatively well in the downstream because the crosstalk gains attenuate with the direct channel gains. The upstream has a near-far effect and IWF is expected to perform rather poorly since it does not use any power back-off strategy.

For ASB, a 1000-m reference line is used in both upstream and downstream cases. The multiple reference lines used for the ASB-MRL algorithms are chosen as the lower bound and the upper bound on the all-CO user's line lengths. That is, a short line of 500m and a long line of 1000m are chosen as the reference lines. Finally, the power back-off factor on the multiple reference lines are provided in Table 4.3. The factors were chosen such that a factor of 10 separates the potentially strong and weak reference lines. In the downstream direction for the all-CO case, both reference lines are equal since their transmitters are both located at the CO causing the direct channel and crosstalk gains to attenuate together. Thus both

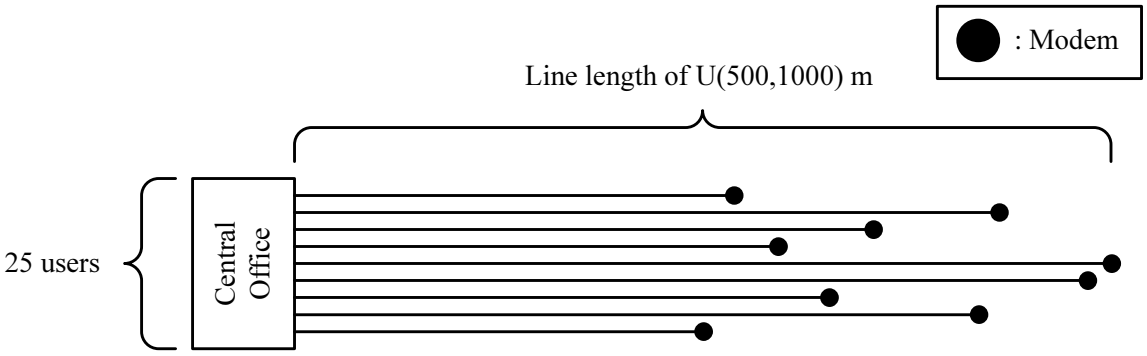


Fig. 4.7 One possible realization of the all-central-office network topology. Each user has a line length taken from a uniform distribution between 500 and 1000m.

Table 4.3 Power back-off factor used by the multiple reference lines in the all-CO network.

direction	500m	1000m
upstream	0.01	1
downstream	1	1

reference lines are allowed to transmit at their full total transmit power. In the upstream case, the near-far effect is present and the 500-m line can generate high amounts of crosstalk onto the 1000-m line. Hence, the stronger 500-m line is only allowed to use 10% of its total transmit power.

The reference line PSDs are generated using water-filling assuming no crosstalk and with the full total transmit power available. The power back-off factor is then applied to the PSD instead of directly on the total transmit power. This is to ensure that the PSD covers as many frequency tones as possible as performing water-filling with tighter power restraints will also likely restrain the PSD reach.

The results for the upstream case are provided in Table 4.4 where the mean and the standard deviation of the sum rate over all 100 network realizations is given. As expected, IWF performs poorly due to the presence of the near-far effect. As for the ASB-MRL algorithms, the results show that ASB-DSB and ASB-SCALE can perform almost as well as DSB and SCALE even if using only two reference lines. Moreover, the ASB-MRL algorithms were able to converge to within 98% in about 4-5 times fewer iterations than the network-level algorithms.

In the downstream case, it was determined that the reference line needing other reference

Table 4.4 Results for the all-CO network for upstream transmission.

Algorithm	Mean sum rate (Mbps)	Standard deviation on the rate (Mbps)	Mean number of iterations to 98%
IWF	145.15	9.30	2
ASB	135.45	23.91	2
DSB	178.14	7.06	9.42
ASB-DSB	177.07	7.17	2
SCALE	178.14	7.06	10.75
ASB-SCALE	177.07	7.17	2.43

Table 4.5 Results for the all-CO network for downstream transmission.

Algorithm	Mean sum rate (Mbps)	Standard deviation on the rate (Mbps)	Mean number of iterations to 98%
IWF	627.92	19.45	2
ASB	631.49	18.45	2
DSB	640.68	19.67	2.76
ASB-DSB	625.89	22.07	2
SCALE	640.72	19.60	17.03
ASB-SCALE	625.92	22.06	3.94

line condition developed in Section 4.5 was not met. In the 500-m reference line's case, the condition for repeating was satisfied on most of the frequency tones. Thus, the number of reference lines was doubled to two 500-m and two 1000-m reference lines. The results for the downstream case are shown in Table 4.5. As expected, IWF performs relatively well and in fact better than ASB-SCALE but not by much.

4.6.2 Mixed-CO-RT Network Simulation

The mixed-CO-RT network has 12 users connected to a CO and 13 users connected to an RT. Each user's line length is taken from a uniform distribution lower bounded at 300m and upper bounded at 600m. The network topology is pictured in Figure 4.8. This kind of network represents the case where an RT is used to reduce the line length of the longer users. However, by bundling the RT and CO together, the near-far effect now appears in both up and downstream directions. Thus, it is expected that IWF will perform poorly in both transmission directions. On the other hand, network-level algorithms will be able to minimize the near-far effect by performing a better PSD allocation.

For ASB, the weakest possible line is chosen as the reference line. Thus, in the upstream

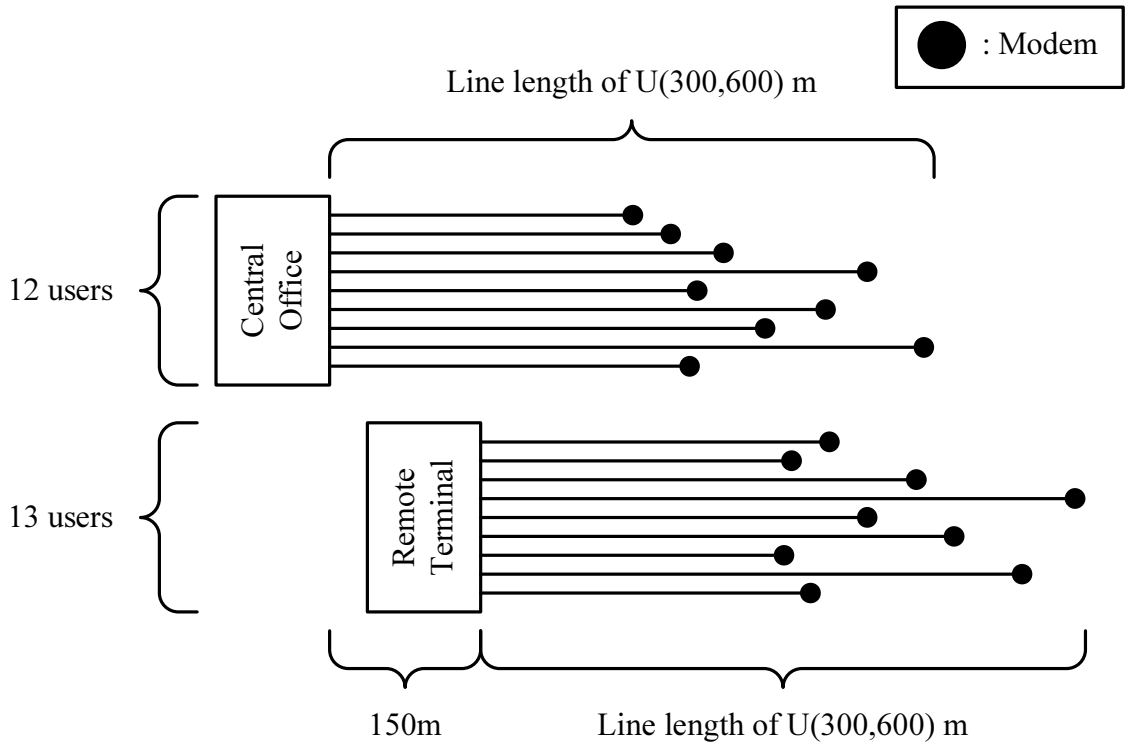


Fig. 4.8 One possible realization of the mixed-central-office-remote-terminal network topology. The distance between the CO and RT is 150m and each user has a line length taken from a uniform distribution between 500 and 1000m.

case, this corresponds to a 600-m RT line. In the downstream case, the 600-m CO line is used. The multiple reference lines used for the ASB-MRL algorithms are chosen as the lower bound and the upper bound on the CO user's line lengths and on the RT user's line length. That is, the virtual network is made up of a CO-300-m line, a CO-600-m line, an RT-300-m line, and an RT-600-m line. The power back-off factors are given in Table 4.6. Like in the all-CO simulations, the factors are chosen to limit the total transmit power of the reference lines with the potential to produce large amount of crosstalk onto the other reference lines. In the downstream direction, the RT lines can be considered as near users relative to the CO lines. Hence, both RT lines have their total transmit power reduced by 10. In the upstream direction, there is the relatively weak RT-600-m line and the relatively strong CO-300-m line with its transmitter located closest to the other reference line's receiver. Each reference line can potentially generated large amount of crosstalk onto the RT-600-m line. Thus, except for the RT-600-m line, the power back-off factor reduces the total

Table 4.6 Power back-off factor used by the reference lines in the mixed-CO-RT network.

direction	CO 300m	CO 600m	RT 300m	RT 600m
upstream	0.01	0.1	0.1	1
downstream	1	1	0.1	0.1

Table 4.7 Results for the mixed-CO-RT network for upstream transmission.

Algorithm	Mean sum rate (Mbps)	Standard deviation on the rate (Mbps)	Mean number of iterations to 98%
IWF	199.23	8.36	2
ASB	128.49	9.18	2
DSB	271.51	6.68	17.83
ASB-DSB	270.65	6.86	2
SCALE	271.51	6.68	19.43
ASB-SCALE	270.65	6.86	3

transmit power by 10 on every line. Similarly, the strong CO-300-m line can produce large amounts of crosstalk onto the two middle lines, the CO-600-m and RT-300-m lines. Thus, its total transmit power is further reduced by 10.

The results for the upstream case is provided in Table 4.7 and for the downstream case in Table 4.8. In both cases, IWF performs poorly as expected due to the near-far effect. ASB performs much worser than IWF which shows that the weakest possible user as the sole reference line is not a good choice as the sole reference line. Moreover, the inability to represent the overall network as a single reference line hinders ASB and may in fact explain why it performs so poorly next to IWF. The mean sum rate obtained by ASB-DSB and ASB-SCALE are near the ones obtained by DSB and SCALE. This shows that the ASB-MRL algorithms can obtain near-optimal results. Moreover, the ASB-MRL algorithms require 5 times fewer iterations for 98% convergence than their network-level counterparts. This shows that, even if DSB or SCALE is implemented on a network, initializing the PSD using the ASB-MRL algorithms can provide significant improvements in terms of convergence speed.

4.7 Conclusion

This chapter compared IWF and DSB to highlight the differences between the IWF and the optimal network-level algorithms from a water-filling viewpoint. The highlights showed

Table 4.8 Results for the mixed-CO-RT network for downstream transmission.

Algorithm	Mean sum rate (Mbps)	Standard deviation on the rate (Mbps)	Mean number of iterations to 98%
IWF	792.59	13.42	2
ASB	612.35	36.23	2
DSB	869.31	18.32	12.24
ASB-DSB	870.90	17.73	2
SCALE	872.08	17.53	29.44
ASB-SCALE	870.94	17.70	3.03

that network-level algorithms produce frequency-selective water-filling levels unlike the flat water-filling level from the user-level IWF. Furthermore, it showed that stronger users must not user their full maximum total transmit power, a process similar to that of Flat PBO. Using these highlights, an extension to IWF through the form of an added Lagrange multiplier offset on the water-filling procedure was introduced as a way of giving IWF the potential to achieve near-optimal performances.

The concept of multiple reference lines was introduced as a method capable of generating the Lagrange offset values. Using the multiple reference lines, two ASB-MRL algorithms were developed: ASB-DSB and ASB-SCALE. A relaxation on the ASB-MRL algorithms was given under the well-represented-network approximation and allowed the use of constant offsets. This has the effect of reducing ASB-MRL's per-iteration complexity from $\mathcal{O}(KN)$ to $\mathcal{O}(K)$, the same as IWF.

The chapter then looked at using the real network's topology to produce the virtual network. From this method, the sum rate versus the iteration number was plotted for a particular network and showed that the ASB-MRL algorithms can converge much faster than their network-level counterparts. Then, instead of going into the many ways of choosing the reference lines to represent a DSL network given what kind of network information is available, this chapter provided three conditions of a good set of reference lines.

Finally, a Monte Carlo style simulation over an all-CO network and a mixed-CO-RT network was performed and showed that the ASB-MRL algorithms can produce near-optimal results by only using a few reference lines that generalized the overall network.

Chapter 5

Conclusion

5.1 Summary

In the first part, this thesis introduced two new methods for performing the water-filling algorithm. The first method relies on projection which is capable of converging to the water-filling solutions in one to K iterations when no PSD mask is present. The GRWF algorithm is the second method and was derived for the generalized water-filling problem where the sum bit-loading and the sum power are weighted. It was shown that both generalized rate adaptive and fixed margin problems result in the same solution and hence GRWF was designed to work using both minimum target rate and maximum total transmit power constraints concurrently. The GRWF algorithm was then shown to have a computational complexity depending only on the number of frequency tones thus allowing to converge within a discrete number of iterations. At the same time, it was shown that its computational load is smaller than the other water-filling methods. Finally, a discrete bit-loading method for the water-filling-like algorithms such as DSB and SCALE was introduced.

In the second part, this thesis concentrated on bridging the gap between user-level and network-level algorithms. Through an analysis of the differences between IWF and network-level algorithms through effective water-filling level comparison, a suitable modification on IWF able to produce network-level results was proposed under the form of a Lagrange multiplier offset variable. The virtual network and the multiple reference lines forming it was proposed as a method for producing the offset variables. From this, a new resource allocation problem over a virtual network was introduced and two Autonomous Spectrum Balancing using Multiple Reference Lines (ASB-MRL) algorithms were developed: ASB-

DSB and ASB-SCALE. A relaxation on the ASB-MRL algorithms in the form of a good virtual network approximation was proposed to allow the use of constant Lagrange offsets. This has the benefit of reducing ASB-MRL's per-iteration computational complexity to that of IWF. Then, three conditions on a good set of reference lines were proposed. The first condition encouraged the reference PSD to cover as many frequency tones as possible in order to prevent localized water-filling. The second condition showed that proper total transmit power restraint on reference lines with large outgoing crosstalk gains is essential. The third condition showed that a virtual network must also be representive of the actual network in terms of crosstalk sources to each reference line. Finally, a Monte Carlo style simulation over two different types of DSL network was performed to show the capabilities of the ASB-MRL algorithms to perform near-optimally while also requiring fewer iterations for convergence than their network-level counterparts.

5.2 Future Research Directions

This section presents potential research directions that this thesis has not yet covered.

In Chapter 3, the discrete bit-loading for water-filling-like algorithms was shown to be better than using bisection on DSB, SCALE, and the ASB-MRL algorithms. However, the discrete bit-loading approach requires a number of iterations that increases with both the number of frequency tones and the number of bit-loading levels. A future research direction is to further reduce the computational load of water-filling-like algorithms.

In Chapter 4, the information contained in the choice of multiple reference lines was used to generate the Lagrange multiplier offset variables used in the modified IWF. Another generation method of potential interest is the use of machine learning to extract any correlation that exists between the measured interferences, channel gains, frequency tone numbers, and Lagrange offsets. Using only this information, no parameters such as the set of reference lines would need to be set at initialization allowing for easier integration into a DSL network.

Also in Chapter 4, a set of criteria on the reference lines was provided in order to have ASB-MRL producing near-optimal results. A future research direction is to create a set of systematic methods for generating the virtual network with each method depending on the amount of available information about the actual network.

Finally, DSL networks may also use vectored transmission in which crosstalk is cancelled

using signal coordination at the central office or at the remote terminal. However, vectoring has a very high computational complexity that grows with the number of vectored users. As a tradeoff, mixtures of vectored and non-vectored have been recently proposed in [43, 44] where some groups of users are vectored together to cancel some crosstalk. It would be an interesting avenue to adapt and apply ASB-MRL over mixtures of vectored and non-vectored networks.

References

[1] ASSIA Inc., "Perfecting the last mile: Advanced DSL management," White Paper, 2010.

[2] *Very High Speed Digital Subscriber Line*, ITU-T Rec. G.993.2 Std., Feb. 2006.

[3] P. Tsiaflakis, Y. Yi, M. Chiang, and M. Moonen, "Green DSL: Energy-efficient DSM," in *2009 IEEE International Conference on Communications (ICC 2009)*, Jun. 2009, pp. 1 – 5.

[4] *ATIS-0600007 Dynamic Spectrum Management Technical Report*, American National Standards Institute Std., May 2007.

[5] K. B. Song, S. T. Chung, G. Ginis, and J. M. Cioffi, "Dynamic spectrum management for next-generation DSL systems," *IEEE Communications Magazine*, vol. 40, no. 10, pp. 101 – 109, Oct. 2002.

[6] K. J. Kerpez, D. L. Waring, S. Galli, J. Dixon, and P. Madon, "Advanced dsl management," *IEEE Communications Magazine*, vol. 41, no. 9, pp. 116 – 123, Sep. 2003.

[7] W. Yu, G. Ginis, and J. M. Cioffi, "Distributed multiuser power control for digital subscriber lines," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 5, pp. 1105 – 1115, Jun. 2002.

[8] R. Cendrillon, F. Liming, J. Chou, G. Long, C. Hung, and D. Wei, "DSM from theory to practice," in *2008 IEEE Global Telecommunications Conference (GLOBECOM 2008)*, Nov./Dec. 2008, pp. 1 – 4.

[9] E. Van den Bogaert, T. Bostoen, J. Van Elsen, R. Cendrillon, and M. Moonen, "DSM in practice: Iterative water-filling implemented on ADSL modems," in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2004)*, May 2004, pp. 337 – 340.

[10] S. Huberman, C. Leung, and T. Le-Ngoc, "Dynamic spectrum management (DSM) algorithms for multi-user xDSL," *IEEE Communications Surveys and Tutorials*, vol. 12, no. 1, to appear in First Quarter 2012.

[11] *T1.417-2001 (Issue 1) Spectrum Management For Loop Transmission Systems*, American National Standard For Telecommunications Std., May 2001.

[12] M. Monteiro, A. Gomes, N. Lindqvist, B. Dortschy, and A. Klautau, "An algorithm for improved stability of dsl networks using spectrum balancing," in *2010 IEEE Global Telecommunications Conference (GLOBECOM 2010)*, Dec. 2010, pp. 1 – 6.

[13] S. Panigrahi, Y. Xu, and T. Le-Ngoc, "Multiuser margin optimization in digital subscriber line (DSL) channels," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1571 – 1580, Aug. 2006.

[14] S. Jagannathan, H. Soo Chan, and J. M. Cioffi, "Margin optimization in digital subscriber lines employing level-1 dynamic spectrum management," in *2008 IEEE International Conference on Communications (ICC 2008)*, May 2008, pp. 441 – 446.

[15] S. Jagannathan, "Interference and outage optimization in multi-user multi-carrier communication systems," Ph.D. dissertation, Stanford University, Jun. 2008.

[16] R. G. Gallager, *Information Theory and Reliable Communication*. New York, NY, USA: John Wiley & Sons, Inc., 1968.

[17] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York, NY, USA: Springer, 2006.

[18] R. Cendrillon, J. Huang, M. Chiang, and M. Moonen, "Autonomous spectrum balacing for digital subscriber lines," *IEEE Transactions on Signal Processing*, vol. 55, no. 8, pp. 4241 – 4257, Aug. 2007.

[19] J. Huang, R. Cendrillon, M. Chiang, and M. Moonen, "Autonomous spectrum balancing (ASB) for frequency selective interference channels," in *2006 IEEE International Symposium on Information Theory (ISIT 2006)*, Jul. 2006, pp. 610 – 614.

[20] P. Tsiaflakis, M. Diehl, and M. Moonen, "Distributed spectrum management algorithms for multiuser DSL networks," *IEEE Transactions on Signal Processing*, vol. 56, no. 10, pp. 4825 – 4843, Oct. 2008.

[21] R. Moraes, B. Dortschy, A. Klautau, and J. Rius i Riu, "Semiblind spectrum balancing for dsl," *IEEE Transactions on Signal Processing*, vol. 58, no. 7, pp. 3717 – 3727, Jul. 2010.

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

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

[24] R. Moraes, B. Dortschy, A. Klautau, R. Zampolo, and J. Rius i Riu, "Optimal solution for the fixed margin problem in digital subscriber lines," in *2008 3rd International Symposium on Communications, Control, and Signal Processing (ISCCSP 2008)*, Mar. 2008, pp. 1395 – 1399.

[25] R. Lui and W. Yu, "Low-complexity near-optimal spectrum balancing for digital subscriber lines," in *2005 IEEE International Conference on Communications (ICC 2005)*, May 2005, pp. 1947 – 1951.

[26] R. Cendrillon and M. Moonen, "Iterative spectrum balancing for digital subscriber lines," in *2005 IEEE International Conference on Communications (ICC 2005)*, May 2005, pp. 1937 – 1941.

[27] M. Chiang, C. W. Tan, D. P. Palomar, D. O'Neill, and D. Julian, "Power control by geometric programming," *IEEE Transactions on Wireless Communications*, vol. 6, no. 7, pp. 2640 – 2651, Jul. 2007.

[28] L. P. Qian and Y. Jun, "Monotonic optimization for non-concave power control in multiuser multicarrier network systems," in *28th Conference on Computer Communications (INFOCOM 2009)*, Apr. 2009, pp. 172 – 180.

[29] L. P. Qian, Y. J. Zhang, and J. Huang, "MAPEL: Achieving global optimality for a non-convex wireless power control problem," *IEEE Transactions on Wireless Communications*, vol. 8, no. 3, pp. 1553 – 1563, Mar. 2009.

[30] Y. Xu, T. Le-Ngoc, and S. Panigrahi, "Global concave minimization for optimal spectrum balancing in multi-user DSL networks," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 2875 – 2885, Jul. 2008.

[31] P. Tsiaflakis, J. Vangorp, M. Moonen, J. Verlinden, and K. V. Acker, "An efficient search algorithm for the lagrange multipliers of optimal spectrum balancing in multi-user xDSL systems," in *2006 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2006)*, May 2006, pp. IV 101 – IV 104.

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

[33] J. Papandriopoulos and J. S. Evans, "SCALE: A low-complexity distributed protocol for spectrum balancing in multiuser DSL networks," *IEEE Transactions on Information Theory*, vol. 55, no. 8, pp. 3711 – 3724, Aug. 2009.

[34] D. D. Yu, K. Seong, and J. M. Cioffi, "Multiuser discrete bit-loading for digital subscriber lines," in *2007 IEEE International Conference on Communications (ICC 2007)*, Jun. 2007, pp. 2755 – 2760.

[35] W. Yu, "Multiuser water-filling in the presence of crosstalk," in *Information Theory and Applications Workshop, 2007*, vol. 1, Jan./Feb. 2007, pp. 414 – 420.

[36] D. Hughes-Hartogs, "Ensemble modem structure for imperfect transmission media," U.S. Patents Nos. 4,679,227 (Jul. 1987), 4,731,816 (Mar. 1988) and 4,833,706 (May 1989).

[37] S. Jagannathan and J. M. Cioffi, "Distributed adaptive bit-loading for spectrum optimization in multi-user multicarrier systems," *Physical Communication*, vol. 1, no. 1, pp. 40 – 59, Mar. 2008.

[38] Y. Xu, T. Le-Ngoc, and S. Panigrahi, "Selective iterative waterfilling for digital subscriber lines," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, 2007, 11 pages.

[39] Y. Xu, "Dynamic resource allocation in multiuser multicarrier fading environments," Ph.D. dissertation, McGill University, Aug. 2007.

[40] E. Altman, K. Avrachenkov, and A. Garnaev, "Closed form solutions for symmetric water filling games," in *27th Conference on Computer Communications (INFOCOM 2008)*, Apr. 2009, pp. 673 – 681.

[41] C. Leung, S. Huberman, and T. Le-Ngoc, "Autonomous spectrum balancing using multiple reference lines for digital subscriber lines," in *2010 IEEE Global Telecommunications Conference (GLOBECOM 2010)*, Dec. 2010.

[42] H. Zou, A. Chowdhery, and J. M. Cioffi, "A centralized multi-level water-filling algorithm for dynamic sprectrum magement," in *2009 Conference Record of the Forty-Third Asilomar Conference on Signals, Systems and Computers*, Nov. 2009, pp. 1101 – 1105.

[43] M. Mohseni, G. Ginis, and J. M. Cioffi, "Dynamic spectrum management for mixtures of vectored and non-vectored DSL systems," in *2010 44th Annual Conference on Information Sciences and Systems (CISS)*, Mar. 2010, pp. 1 – 6.

[44] A. Chowdhery and J. M. Cioffi, "Dynamic spectrum management for upstream mixtures of vectored & non-vectored DSL," in *2010 IEEE Global Telecommunications Conference (GLOBECOM 2010)*, Dec. 2010.