A Power Interval Perspective on Additive White Gaussian Noise (AWGN) Channels

by

Li Shu

B.E. (1992) and M.E. (1994), Electrical Engineering
The Cooper Union for the Advancement of Science and Art

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical Engineering and Computer Science

at the

Massachusetts Institute of Technology

February 2000

Author ....
Department of Electrical Engineering and Computer Science
January 30, 2000

Certified by ....
Robert G. Gallager
Professor of Electrical Engineering
Thesis Supervisor

Accepted by ....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

A Power Interval Perspective on Additive White Gaussian Noise (AWGN) Channels

by

Li Shu

Submitted to the Department of Electrical Engineering and Computer Science
on January 30, 2000, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

We present a new perspective on additive white Gaussian noise (AWGN) channels that separates user and channel attributes. Following this separation, various questions concerning achievability and successive decoding can be reformulated as properties of the set of user attributes, which can be determined independently of the actual channel noise. To obtain these properties directly, we introduce a graphical framework called the *power diagram.* Based on graphical manipulations in this framework, our results on N-user multi-access channels include the following:

1. simplifying the achievability condition to an algorithm requiring $O(N \ln N)$ computations

2. simplifying the check of whether a given rate tuple is decodable with simple successive decoding (to be defined) to an algorithm requiring $O(N \ln N)$ computations

3. developing a technique for *power-reduced successive decoding*, accompanied by the set of rate tuples for which such a technique is applicable, and an algorithm that checks whether a given rate tuple is decodable with this technique requiring $O(N \ln N)$ computations

4. presenting a class of graphical constructions for splitting any achievable rate tuple into a set of virtual users that allows successive decoding. These constructions deal with rate tuples not on the dominant face in a natual way, whereas previous works have viewed these rate tuples as a somewhat *ad hoc* extension of the dominant face results

5. presenting a class of graphical constructions that facilitate successive decoding to any achievable rate tuple using the *time-sharing* technique, improving the known upper bound on decoding complexity (using this combination of techniques) to $2N - 1$

Thesis Supervisor: Robert G. Gallager
Title: Professor of Electrical Engineering

Acknowledgments

First of all, let me thank my thesis supervisor, Prof. Robert G. Gallager, for patiently supporting and advising me through my graduate career. I have come to MIT wanting to solidify my background in theory and to acquire more tools for analysis and understanding. I have learned a lot, and it has been a mind-opening experience.

I would also like to take the time to thank my committee, which includes Profs. Peter Elias, Dave Forney, and Sanjoy Mitter, for patiently going through my writing and providing many valuable suggestions.

My interest in engineering can be traced all the way back to childhood. I remember my mother teaching me oragami just after I learned to talk and walk. Over time, my parents encouraged me to think creatively when facing a problem, and, perhaps most importantly, instilled in me a sense of pride for working with both hands and mind. I will always remember your teaching on "things are dead, but people are alive".

I would also like to thank my American parents, Mr. and Mrs. Durward C. and Kay S. Ellis, for providing a loving and nurturing environment during some of the most difficult and dangerous times of my life, and for helping to shape my character and mind during the formative years. I will follow your teachings that "if I saw something I don't like, to make sure to do it differently when I am in that position".

I had thought that something in me is not fit for romantic relationships until I found my love merely a year ago. She makes me laugh, and helps me to see from a larger perspective when I become too focused and intense. With her, I am no longer a lost and wandering soul. Darling, I am so happy that we will be spending the rest of our lives together. To her parents: thank you, thank you, and THANK YOU for bring her up the way she is. I look forward to learning many things from you, chief among which is how you brought up your children.

My life has been made up of a sequence of steps. I have been extremely fortunate to have received timely and selfless help and guidance during many of these steps, particularly during those when important (sometimes critical) decisions are required. Over time, a number of you have become my lifelong mentor, whose considerations and opinion I have come to depend upon. I have strived to make best use of my gifts, received from my parents and you, and hope that, in some meaningful way, I have been able to make your generosity towards and efforts for me worthwhile, so that you may be proud of your associations with me.

I would like to express my profound gratitude to the mentors in my life. They include,

- Mr. Bernier Laster Mayo (yeah! I remembered your middle name!), headmaster of the St. Johnsbury Academy, who played an instrumental role during my transition into the New

World, and whose confidence in me (which I know I don't deserve) continue to inspire me to strive forward

- Mr. Brett P. Hoffman of Newark, Vermont, who helped me in ways and means far surpassed the duty and the responsibility of an academic advisor during my year at the St. Johnsbury Academy, and continue to be my mentor today

- Dr. Alan Graham, of formally St. Johnsbury, Vermont, and now of Boulder, Colorado, whose keen insight and adroitness in introducing critical understandings have helped me to avoid pitfalls in a number of occasions, and guides me onto ultimately living a balanced life

- Dean Richard Bory, dean of admissions, and Dean Stephen Baker, dean of students, both at Cooper Union, (My Deans!), who saw me as what I could be, and have supported and encouraged me throughout my college and post-graduate studies

- Prof. Fred F. Fontaine of EE at Cooper Union, without whose indulgence and support in my pursuing everything possible, and foresight to stop me just before I had "enough ropes to hang myself on", I would not be where I am today

Thanks also goes to

- Mrs. GuoFang He of the 10^{th} Middle School of Nanjing, who was my class mentor there

- my Chinese teach at 10^{th} grade at the 10^{th} Middle School of Nanjing, whose name I have unfortunately forgotten though I remember she was also a reporter for the local newspaper

- Dr. Shu Zheng of Nanjing Normal University

- (the late) Mr. Paul Sanderson, founder of the ASSIST program which brought me over to the United States

- Prof. Campbell Searle of EECS at MIT, and Mr. J. Lawrence Peirson (Jake), dean of the WHOI-MIT joint program, who bestowed confidence and trust in me during moments of doubt

- Prof. Mitch Trott, whose enthusiastic champion of others' (particularly "lowly" students') ideas and work sets a great example for me

- Drs. Subramanium Rajan and DaJun Tang of the Woods Hole Oceanographic Institution (WHOI), who selflessly supported me to pursue what is the best for me

- Dr. Yehuda Avniel, whom I met during his brief visit to LIDS a few years ago, and who had guided me through my graduate studies as a caring and experienced friend with patience, understanding, and foresight
- Dr. Robert Bruen, the new system administrator at LIDS with a Ph.D. degree(!), whose judgments and opinion I have started to depend upon increasingly, even though we have not know each other very long
- my brother, Han, whose overflowing enthusiasm is contagious, and whose caring and understanding I can always depend upon
- my (adopted) little sister Euree Kim, who walked by my side through times of doubt and draught
- LIDS administrators Nancy Young-Wearly, Maria J. (Fifa) Monserrate, Kate O'Sullivan, and Doris Lannoy Inslee, EECS (course VI) graduate office administrators Monica Bell, Marilyn Pierce, and Peggy Carney, whose timely guidance have made navigating through the un-charted waters (for me) of graduate studies much easier

In addition to a number of individuals mentioned in the above, my extended family also includes my maternal grandmother Mrs. AiZhen Wang (whose name I learned because I was "obnoxiously" insistent), Drs. XinYe Cherry Lei and YongRen Benjamin Peng and their children April + Victor + Felicia, Claudiu Vaduva, Prof. Lui Lam, auntie WanHua Xu, Li Shen, Mike + Cathy Ellis (Great catch, Mike!), Jonathan + Christopher + Esther + Larry Denham, Marian + Julian Butler, and Dr. & Mrs. Phil & Wendy Lin + little Corisa (Please forgive me if I have forgotten someone here – it is a rather large extended family.). Besides forming the back-bone that inspires on-ward, they have also provided many valuable feedbacks, suggestions, and insights at various junctures of my life. In almost every sense possible, I would not have gotten here if it were not for these people. Thank you.

I want to take a moment to remember my (late) maternal grandfather Mr. ChengLiang Xu, (late) uncle Mr. GuangHua Xu, (late) great uncle and auntie Mr. & Mrs. FuLiang Xu (I don't even know her name!), (late) auntie Mrs. YuHua Xu, for their care and love. How I wish you could all be here to share this with me!

Finally, I would like to dedicate this thesis to the many individuals out there whose natural gifts far surpass mine, who developed splendid ability, personality, and maturity despite the fact that they were not able to attend an institution of their choice, by whom I have been humbled in so many ways – I look forward to meeting many more of you in the future, and this is for you.

Contents

List of Figures

Chapter 1

Introduction

In his landmark work in 1948 [13], C.E. Shannon proposed a mathematical theory for modern telecommunication systems which we now call *information theory*. Among other things, he clarified and defined a number of critical concepts such as *communication channel*, *information measure*, *reliable communication*, and *channel capacity*. These concepts were later expanded into modern information theory (c.f. [7], [6]), and continue to define the standards of usage today.

To illustrate his theory, Shannon presented a study of an additive white Gaussian noise (AWGN) channel model with one transmitter and one receiver. This channel model is a simple abstraction of the most basic problem of transmitting data in the presence of noise. We refer to this channel here as the *AWGN single-user channel* model, to distinguish it from other models with more than one transmitter. The effect of this channel is to distort any signal passing through it by adding a white Gaussian noise signal of a known spectral density.

Using the concept of sufficient statistics, the effect of adding white Gaussian noise to a signal low-pass band-limited to W hertz per second can be equivalently represented by a discretization to a set of low-pass band-limited basis signals. We can capture the most important aspect of the AWGN single-user channel by a *discrete time AWGN single-user channel model* in which, during each interval of $\frac{1}{2W}$ seconds, the transmitter uses the channel once, and the receiver makes a single observation of the sum of the transmission $X[k]$ and the channel noise $Z[k]$. The channel noise $\{Z[k]\}$ is assumed to be a set of independent identically distributed (i.i.d) zero mean Gaussian random variables of variance N_o. In terms of the white Gaussian noise variances, n_o is the spectral density of the noise (often denoted as $\frac{N_o}{2}$). For simplicity, we illustrate this channel model as in Figure 1-1.

From Nyquist's theorem, the transmission can be represented as a linear combination of $2W$ basis signals per unit time. One may equivalently regard this channel as one in which a

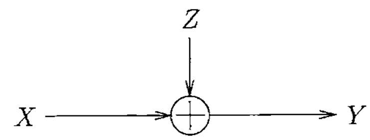


Figure 1-1: The discrete time AWGN single-user channel

single number (one degree of freedom in the original band-limited signal) is transmitted every $\frac{1}{2W}$ seconds. The capacity of the band-limited channel in natural units (nats) per degree of freedom (i.e. per interval $\frac{1}{2W}$) was shown by Shannon to be

$$\frac{1}{2}\ln\left(1+\frac{p}{n_o}\right) \tag{1.1}$$

where $p = p_W/2W$ is the allowable signal energy per dimension, which we call the *power constraint.* This is Shannon's famous capacity formula for Gaussian noise.

In the rest of this document, unless otherwise indicated, the units of power constraint and transmission rate are all per dimension.

In this discrete-time AWGN single-user channel, the receiver may decode the received message using hypothesis testing, where each possible message that the transmitter may send is one hypothesis. The transmission is decoded to (decided to be) one of the possible messages (hypotheses) according to certain criteria. We say an error has occurred in decoding if the transmitted message and the decoded message are not the same.

In his original work, Shannon defined *reliable communication* as the situation when the probability of decoding error can be made arbitrarily small. In addition, he showed that using (redundancy) coding and the maximum *a posteriori* (MAP) decision rule, the decoding error averaged over an appropriate probabilistic ensemble of codes can be made arbitrarily small for every transmission rate less than the channel capacity, and can not be made arbitrarily small for rates above that.

We now say that a rate tuple is *achievable* in a communication channel if reliable communication can be facilitated for every rate tuple that it component-wise dominates strictly, and we call the set of achievable rate tuples in a given channel the *achievable rate region* of the channel. Observe, for example, in the AWGN single-user channel above, a transmission rate r is achievable in the given channel if and only if

$$r < \frac{1}{2}\ln\left(1+\frac{p}{n_o}\right) \tag{1.2}$$

The achievable rate region of this channel is the closed interval $[0, \frac{1}{2}\ln\left(1+\frac{p}{n_o}\right)]$.

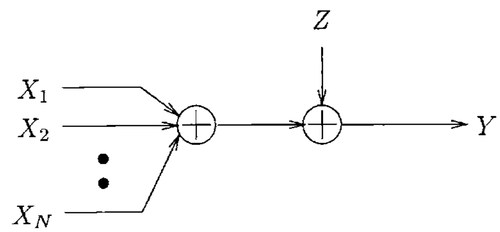


Figure 1-2: The discrete time AWGN multi-access channel

In time, several generalizations of the single-user channel model came into existence, one of which has been termed the *multi-access* channel model. A multi-access channel consists of multiple transmitters, each driven by an independent source, and a single receiver. Modern cellular telephony is based on the multi-access channel model.

When the noise in the multi-access channel is AWGN, we may use the same argument as in the AWGN single-user channel to discretize the model. Figure 1-2 illustrates the *discrete time AWGN multi-access* channel model.

In the early 1970s, the achievable rate region of the multi-access channel was derived by Liao [11], and Ahlswede [1] independently. In an N-user AWGN multi-access channels, let p_i and r_i denote the power constraint and the transmission rate of user i (transmission X_i) for $i \in [1, \ldots, N]$. They showed that the achievable rate region is the set of rate tuples satisfying

$$0 \leq \sum_{i \in \mathcal{S}} r_i \leq \frac{1}{2} \ln \left(1 + \frac{\sum_{i \in \mathcal{S}} p_i}{n_o} \right), \forall \mathcal{S} \subseteq [1, \ldots, N] \tag{1.3}$$

where n_o is the noise variance in the channel. When used for determining whether a given rate tuple is in the achievable rate region (and therefore achievable in the given channel), this set of conditions is often referred to as the *multi-access achievability conditions.*

One interpretation of the fact that no rate tuple outside of this region is achievable is offered by Gallager [8]. Specifically, if a super user is able to coordinate the transmission of all users in a subset $\mathcal{S}$, and tell the remaining users to cease their transmissions, then this super user should not be able to transmit beyond its single-user capacity with the combined transmission power of all users in $\mathcal{S}$.

Figure 1-3 illustrates this region for $N = 2$. Observe that this region is bounded by three constraints. Note that every rate tuple in this region is component-wise dominated by at least one rate tuple on the segment of the boundary contributed by the condition in (1.3) for $\mathcal{S} = \{1, 2\}$ (segment AB). For this reason, this segment have been called the *dominant face* of the achievable rate region. In general, the dominant face of an AWGN achievable rate region is the section of the bounding hyper-plane contributed by the condition in (1.3) corresponding

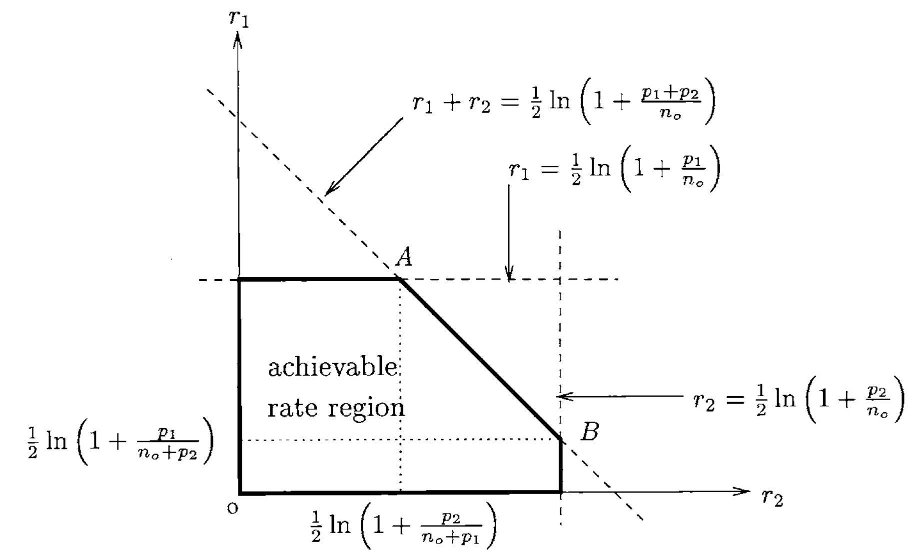


Figure 1-3: The achievable rate region of a 2-user AWGN multi-access channel

to $\mathcal{S} = [1, \dots, N]$.

Moreover, notice that there are 2^N distinct subsets of $[1, \dots, N]$, each (except the empty set) contributing a boundary to the achievable rate region of an N-user AWGN multi-access channel. Hence, checking the achievability of a rate tuple in this channel, using these multi-access achievability conditions directly, requires $O(2^N)$ computations.

Finally, since the AWGN multi-access achievable rate region is bounded by a set of hyper-planes and contains only rate tuples with finite sums, it must be the convex hull of a set of finite number of rate tuples. In particular, the dominant face must also be the convex hull of a set of a finite number of rate tuples. These rate tuples are called the corner points on the dominant face. Note that points A and B are the two corner points on the dominant face of the two-user AWGN multi-access achievable rate region.

One possible approach to decoding a multi-access channel is the technique of *joint decoding.* This decoding technique combines all users' messages into a joint message, and decodes this joint message as if it is transmitted through a single-user channel. Observe that if each user chooses one of K messages to transmit, there are K^N possible joint messages. Choosing the most likely joint message among K^N possible messages, in general, requires $O(K^N)$ computations.

This combination of the computational burden of the multi-access achievability conditions and joint decoding seems to suggest that the complexity of a multi-access channel grows exponentially with the number of users in the channel. The exponentially growing computational burden makes implementation of joint decoding difficult in all but the simplest cases.

To circumvent the difficulty in implementing joint decoding, decoding techniques used in single-user channels have been extended into techniques such as *independent decoding, (naive)*

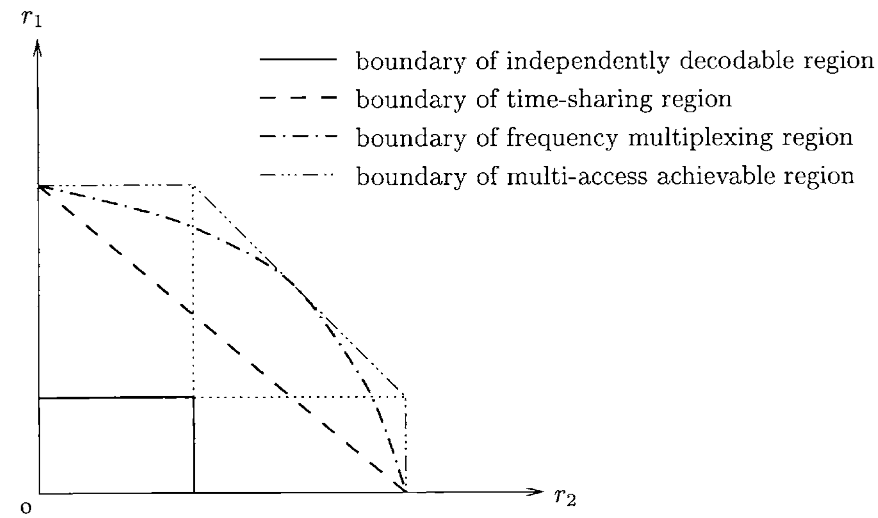


Figure 1-4: Illustrating the rate regions for which independent decoding, time-sharing, and frequency multiplexing techniques may be used

time-sharing, and *frequency multiplexing* in multi-access channels (c.f. [3]). Briefly, independent decoding decodes the users' transmissions one at a time while regarding the transmission of all other users as part of the channel noise; time-sharing allows only one user to transmit during each unit of time (degree of freedom); and frequency multiplexing divides the transmission bandwidth into slots, and allows only one user to transmit within each slot. The important difference between time-sharing and frequency multiplexing is that the latter imposes an average power constraint, whereas the former imposes a (instantaneous) power constraint on each transmission (degree of freedom).

While requiring much less computation than joint decoding, these decoding techniques are unable to achieve all rate tuples in the achievable rate region. Figure 1-4 illustrates the comparison between the rate tuples for which these techniques may be used. However, lacking better alternatives, these techniques have been implemented in practical multi-access communication systems. Most notably, independent decoding and time-sharing techniques are used respectively in IS-95 and GSM, the two most widely-used cellular telephone standard today.

In 1996, Rimoldi and Urbanke [12] showed that, by combining *successive decoding* and *user-splitting*, it is possible to decode every achievable rate tuple in the AWGN multi-access achievable rate region with the computational burden for decoding growing no faster than linearly with the number of users in the channel. Subsequently, I presented a new perspective on AWGN multi-access channels in 1997 [16]. Using a graphical framework called the *power diagram*, I derived a set of equivalent conditions to the AWGN multi-access achievability condition which requires $O(N \ln N)$ computation. The combination of these two results effectively showed that, in theory, an AWGN multi-access channel with N users is no more complex than a linear multiple of N AWGN single-user channels with an ordering. Compared to the original set of multi-access achievability conditions and the joint decoding technique, these new results represent dramatic simplifications for AWGN multi-access channels.

Since then, Yeh and Gallager [18] explored the combination of *successive decoding* and *time-sharing*, and showed that it is possible to decode every achievable rate tuple in the AWGN multi-access achievable rate region with $O(N + \frac{1}{2}N\ln(N))$ computational burden[1]. And Grant et al [9] showed that the combination of *successive decoding* and *user-splitting* may be used in general discrete memoryless multi-access channels to decode every achievable rate tuple in their achievable rate region with an $O(N)$ computational burden.

In this thesis, the perspective that I presented in [16] is formalized to allow additional questions concerning the AWGN multi-access channels to be formulated independent of the actual noise variance in the channel. The power diagram of [16] is extended here to develop solutions to these questions requiring near-linear growth of the computational burden with the number of users in the system. In particular, in addition to simplifying the AWGN multi-access achievability conditions, we use the power diagram to study four combination techniques with successive decoding. They are *simple successive decoding*, *power-reduced successive decoding*, *user-splitting successive decoding*, and *time-shared successive decoding*.

While the set of rate tuples that can be decoded with simple successive decoding (to be defined) is well known, a brute force check of whether a given rate tuple can be decoded with simple successive decoding may require up to $O(2^N)$ computations. Using the power diagram framework, we are able to develop an $O(N \ln N)$ alternative to this check.

As its name suggests, the power-reduced successive decoding technique allows individual users to transmit below their power constraint to facilitate better reception of the other users' transmissions. We propose this combination of techniques here because the required computational burden for decoding is identical to simple successive decoding, which is considerably less than those of the remaining two combinations of techniques. While there are many more rate tuples that can be decoded with power-reduced successive decoding than those with simple successive decoding, this combination can not be used for all achievable rate tuples. We develop a complete description of the set of rate tuples for which this technique is applicable, and an $O(N \ln N)$ algorithm that checks whether a given rate tuple belongs to this set.

User-splitting successive decoding and time-shared successive decoding are discussed respectively in [12] and [18]. For the former, we expand the class of constructions presented in [16] that require no greater decoding complexity than those in [12]. Using the power diagram, we

[1] As we approach the completion of this thesis, we also become aware of a work recently submitted by Rimoldi for publication that reduces the number of successive decoding steps to $O(N)$ with time-sharing.

show that all possible such constructions can be produced by a single graphical construction algorithm. And we present individual constructions other than those in [12] that have various desirable qualities for implementation in practical systems.

We extend the power diagram framework to the *time-power diagram* to study time-shared successive decoding. Using this tool, we improve the results in [18] by reducing the computational requirement further to $O(N)$. This places this technique on an equal footing with the user-splitting technique.

These studies provide theoretical insights into multi-access communication practice, and bring us closer to multi-access communication systems with low complexity and very high efficiency. However, these analysis are performed in theory, which assumes that the transmission of each user may be decoded to arbitrarily small probability of error. This assumption makes it possible to "cleanly" strip (or take "complete" account of) all decoded transmissions for the next successive decoding step. However, the probability of decoding error in real communication systems can not be made arbitrarily small (because of either the length of transmission codes or the constraint on the transmission delays). In other words, the decoded transmissions may not be "cleanly" stripped in practice, and may thus cause additional errors in subsequent successive decoding steps. This phenomenon has been termed *error propagation*. We hope that this remaining roadblock will be better understood, and subsequently removed in the near future.

Because of the diverse nature of the topics covered in this work, more detailed introductions to each topic are presented at the beginning of each chapter.

The rest of this document is organized as follows: in Chapter 2, we introduce the new perspective to the AWGN multi-access channels and the power diagram, and the simplification of the AWGN multi-access achievability conditions; then, various successive decoding methods are studied in Chapters 3 through 5 using the perspective developed in Chapter 2. In particular, simple successive decoding and power-reduced successive decoding are studied in Chapter 3; the combination of user-splitting and successive decoding techniques is studied in Chapter 4; and the combination of time-sharing and successive decoding techniques is studied in Chapter 5. Finally, in Chapter 6, we make a few concluding remarks on the results of our study, and on further application of our perspective to other aspects of the AWGN channels.

Chapter 2

The Power Diagram and Achievability in AWGN Multi-Access channels

This chapter introduces a new perspective on discrete time additive white Gaussian noise (AWGN) multi-access channels, and a power diagram framework to take advantage of this new perspective. We show a graphical approach to the simplification of the AWGN multi-access achievability condition using these concepts.

For simplicity of notation, we define $\mathsf{C}(p, n_o)$ to be the channel capacity of an AWGN single-user channel with channel noise variance n_o and power constraint p, i.e.

$$\mathsf{C}(p, n_o) = \frac{1}{2} \ln \left(1 + \frac{p}{n_o}\right) \tag{2.1}$$

An AWGN multi-access channel is specified by the variance of the additive white Gaussian noise in the channel, a set of desired transmission rates of the users, and a set of power constraints on the users' transmissions. Traditionally, the power constraints are often associated with the channel, and users' desired transmission rates are often considered as the attributes of the users.

One reason for such an association is the definition of channel capacity. To illustrate this point, let n_o denote the noise variance in an AWGN single-user channel, and p the power constraint on the user's transmission. The channel capacity is dependent on the maximal signal-to-noise ratio of the user's transmission in the channel under the transmission power constraint as in (2.1). Reliable communication can be achieved only if the user's desired transmission rate

r is below the channel capacity, i.e.

$$r < \mathsf{C}(p, n_o) \tag{2.2}$$

In an N-user AWGN multi-access channel, let p_i denote the power constraint on user i's transmission, and r_i denote user i's desired transmission rate, for $i \in [1, \ldots, N]$. Under this association of attributes, this AWGN multi-access channel is specified by $N + 1$ parameters, the channel noise variance n_o and power constraints p_i for each $i \in [1, \ldots, N]$. Hence, whether reliable communication can be achieved for these users' rates in this channel is determined by comparing the rate specifications with properties of the channel specifications. Recall, from Chapter 1, that the required comparisons are

$$\sum_{i \in \mathcal{S}} r_i < \frac{1}{2} \ln \left(1 + \sum_{i \in \mathcal{S}} \frac{p_i}{n_o} \right) = \mathsf{C} \left(\sum_{i \in \mathcal{S}} p_i, n_o \right), \quad \forall \mathcal{S} \subseteq [1, \ldots, N] \tag{2.3}$$

Observe that under this association of attributes, introducing a new user into the system increases the dimensionality of the channel specification (hence the dimensionality of the problem) by 1. From this point of view, the fact that the number of computations required to check the achievability of an N-user AWGN multi-access channel grows expontially with N seems to be un-escapeable. Therefore, it would seem that simplification of the AWGN multi-access achievability conditions must start from a new association of these attributes. This is the starting point of our work.

Consider the following association of the attributes in an AWGN channel:

- We regard the noise variance as the only characteristic of an AWGN channel.
- We associate both the power constraint p and the desired transmission rate r with individual users, and hence each user possesses a pair of attributes. For brevity, we denote such a user specifications by (p, r).

The immediate benefit of this new association is that introducing additional users into the system no longer increases the dimensionality of the channel specification. In fact, under this association, the only difference between two AWGN channels, whether single-user or multi-user, is their channel noise variance.

Additionally, this new association avails a new perspective in which properties of sets of user specifications $\{(p_i, r_i), i \in [1, \ldots, N]\}$ may be studied *independently* of the channel noise. For example, since reducing the noise variance in the channel only eases reception of all the users' transmissions, we may consider the following question what is the *maximal noise threshold* in an AWGN channel at which a given set of user specifications can be achieved?

To illustrate, consider an AWGN single-user channel with a user specified by (p, r). Observe that the transmission rate is the capacity of one AWGN single-user channel with noise variance η under the power constraint p, i.e.

$$r = \mathsf{C}(p, \eta) = \frac{1}{2} \ln \left(1 + \frac{p}{\eta}\right) \tag{2.4}$$

Specifically, η is given by

$$\eta = \frac{p}{\exp(2r) - 1} \tag{2.5}$$

Since the capacity of the AWGN single-user channel decreases with increasing channel noise variance, the transmission rate r cannot be achieved under the transmission power constraint p, if the noise variance exceeds η. In other words, η is the *maximal noise threshold* for the user specification (p, r), and we conclude that the given user's power and rate specifications are achievable in the given channel if and only if

$$n_o < \eta \tag{2.6}$$

Comparing to the conventional check of the AWGN single-user achievability as in (2.2), the new check merely reverses the order of the involvement of the channel noise variance and the transmission rate, and thus postpones the involvement of the channel noise variance. While this makes no material difference in the single-user achievability, it facilitates our simplification of the AWGN multi-access achievability conditions. We discuss this presently.

Using the above notation, for each $\mathcal{S} \subseteq [1, \dots, N]$, let

$$\eta_{\mathcal{S}} = \frac{\sum_{i \in \mathcal{S}} p_i}{\exp\left(2 \sum_{i \in \mathcal{S}} r_i\right) - 1} \tag{2.7}$$

By (2.5), $\eta_{\mathcal{S}}$ is the *maximal noise threshold* of the user with the combined rate specifications and power constraints of users in $\mathcal{S}$, i.e. a user with specifications $\left(\sum_{i \in \mathcal{S}} p_i, \sum_{i \in \mathcal{S}} r_i\right)$. By the monotonicity of the logarithm, conditions in (2.3) can be alternatively stated as

$$n_o < \frac{\sum_{i \in \mathcal{S}} p_i}{\exp\left(2 \sum_{i \in \mathcal{S}} r_i\right) - 1}, \quad \forall \mathcal{S} \subseteq [1, \dots, N] \tag{2.8}$$

Substituting (2.7) into (2.8), the set of user specifications is achievable in the given channel if and only if

$$n_o < \min_{\mathcal{S} \subseteq [1,\dots,N]} \eta_{\mathcal{S}} \tag{2.9}$$

From (2.7), notice that each $\eta_{\mathcal{S}}$ is only dependent on the specifications (the desired transmission rates and the power constraints) of users in $\mathcal{S}$. Their minimum is therefore a characteristic of the set of user specifications. In particular, since this set of user specifications is achievable in a given channel if and only if the channel's noise variance n_o is less than the this minimum, we have the following:

Definition 2.0.1. The *maximal noise threshold* of a set of user specifications $\{(p_i, r_i), i \in [1, \dots, N]\}$ is

$$\min_{\mathcal{S} \subseteq [1,\dots,N]} \eta_{\mathcal{S}} \tag{2.10}$$

where $\eta_{\mathcal{S}}$ is given by (2.7).

The discussion above provides the following theorem.

Theorem 2.0.1. *A set of user specifications is achievable in an AWGN multi-access channel if and only if the channel noise variance is no greater than the maximal noise threshold of the set.*

Though only a re-arrangement of the conditions in (2.3), the new formulation in (2.9) makes it possible to first consider the set of user specifications independent of the channel attribute. The involvement of the actual channel attribute, i.e. the noise variance, is postponed until after the appropriate property of the set of user specifications, i.e. its *maximal noise threshold*, is extracted.

While a direct implementation of this new formulation requires a similar number of comparison operations as the original achievability condition, we will see that a structure[1] exists among the set of $\eta_{\mathcal{S}}$ which reduces the computational burden of finding their minimum to $O(N \ln N)$. As will be seen, the most complex part of calculating the minimum in (2.10) turns out to be ordering a certain attribute of the users. In the succeeding chapters, we will focus on simplifying the decoding aspects of the AWGN multi-access channel.

In the rest of this chapter, we will introduce a new framework for studying properties of a set of user specifications such as the *maximal noise threshold* independent of channel noise.

[1]Concurrent to the development of this chapter (as published in [16]), Hanly and Tse [10] showed that the set of user specifications $\{(p_i, r_i), i \in [1, \dots, N]\}$ is a polymatroid with set function given by (2.5), and exhibited an $O(N \ln N)$ solution to the AWGN multi-access achievability.

It is a graphical framework in which both the user specifications and the channel noise may be represented and manipulated. We call it the *power diagram.* Then, our simplification of the AWGN multi-access achievability conditions is developed using this power diagram in the remainder of this chapter. In the succeeding chapters, we will use this new perspective and the power diagram framework to study various capacity-achieving techniques in the AWGN multi-access channel.

The next section introduces the power diagram framework along with a visualization tool for interpreting manipulations within it. Since nearly all key insights discussed in the rest of this document come from the power diagram framework, this visualization tool will be used frequently. We also formulate the AWGN multi-access achievability conditions using this framework in this section. Then, in Section 2.2, the low complexity solutions to the maximal noise threshold and the achievability of a two-user AWGN multi-access channel are derived. The two-user case is simpler than the general N-user case, but embodies the essential ideas. These results are then generalized to the arbitrary N-user case in Section 2.3. Finally, a few comments are offered to prepare the readers for the rest of the document.

2.1 Introducing the Power Diagram

In this section, we first define the power diagram framework and its graphical representation. Then, we formulate the AWGN multi-access achievability conditions in this framework to prepare for its simplification in the rest of this chapter.

The components of the power diagram may be easily seen with the single-user case. From the discussion on AWGN single-user channel in the above, observe that the monotonicity of the logarithm in (2.5) dictates that any two of the three parameters (i.e. the power constraint p, the transmission rate r and the maximal noise threshold η) in the equation uniquely determine the third. Thus, specifying the p and η pair is equivalent to specifying the (p, r) pair. The following definition is based on this equivalence.

Definition 2.1.1. Let a user be specified by its power constraint p and maximal noise threshold η. Then, the user's *power interval* is defined to be the left-open-right-closed interval $(\eta, \eta + p]$ on the real axis.

The real axis on which power intervals are placed is called the *power axis.*

Notice that the lower boundary of a power interval is the user's maximal noise threshold η, the length of a power interval is the user's power constraint p, and one-half the logarithm of the ratio of the two boundaries of a power interval is the user's transmission rate specification, i.e.

$$r = \mathsf{C}(p, \eta) = \frac{1}{2}\left(\ln(\eta + p) - \ln(\eta)\right) \tag{2.11}$$

For convenience, we define the following functions.

Definition 2.1.2. Given a user's power interval $T = (a, b]$,

- $\eta(T) \equiv a$ is the lower boundary of the interval, which, by definition of power interval, is also the maximal noise threshold of the user.
- $\mathsf{p}(T) \equiv \mathsf{p}((a, b]) = b - a$ is the length of the interval, which is the user's power constraint.
- $\mathsf{r}(T) \equiv \mathsf{C}(\mathsf{p}(T), \eta(T)) = \frac{1}{2}\left(\ln(\eta(T) + \mathsf{p}(T))) - \ln(\eta(T))\right)$ is the user's transmission rate specification.

In the following, we place the power intervals of multiple users on a single power axis to form the *power diagram*. The physical sense of such a construct may be discerned from the following graphical relations of the two representations of a user's attributes, i.e. its power and rate specifications and its power interval.

Given a user power interval T, notice from (2.11) that the user's transmission rate specification can also be interpreted as the height span of function $\frac{1}{2}\ln(x)$ for x between the two boundaries of T. In other words, the power constraint and transmission rate of the user specified by a power interval T is given by the dimensions of the two-dimensional vector with both ends on the $\frac{1}{2}\ln(x)$ curve that spans T on the x-axis. Conversely, the power constraint and the transmission rate of a user specify the dimensions of a chord on the $\frac{1}{2}\ln(x)$ curve, and the horizontal projection of this chord is the power interval of the user. The uniqueness of such chords is provided by the uniqueness of the power interval, which is in turn the consequence of the monotonicity and concavity of the $\frac{1}{2}\ln(x)$ function. The existence of such a chord is seen by the additional facts that

$$\lim_{x\to 0} \frac{1}{2}\ln(x) = -\infty \tag{2.12}$$

$$\lim_{x\to \infty} \frac{1}{2}\ln(x) = \infty \tag{2.13}$$

$$\lim_{x\to 0} \frac{d}{dx}\frac{1}{2}\ln(x) = \infty \tag{2.14}$$

$$\lim_{x\to \infty} \frac{d}{dx}\frac{1}{2}\ln(x) = 0 \tag{2.15}$$

These facts are depicted in Figure 2-1.

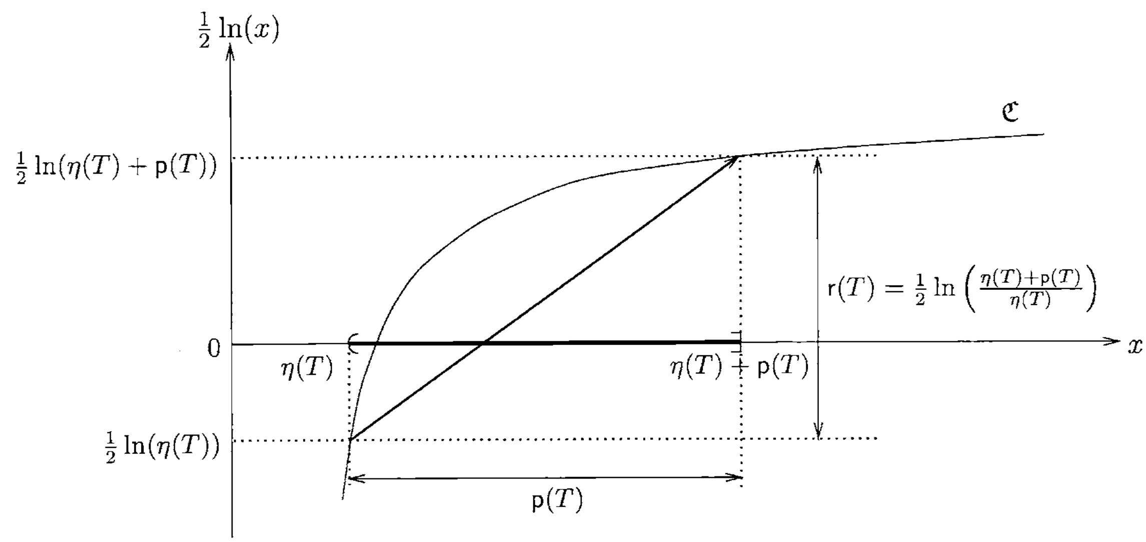


Figure 2-1: Interpreting the power interval representation

Moreover, observe that the same $\frac{1}{2}\ln(x)$ function may be used for interpreting (computing) the transmission rate of any power interval. In this sense, this x-axis and the $\frac{1}{2}\ln(x)$ function is universal to all power intervals[2]. We abstract this universal x-axis to be the power axis in the power diagram. We may thus place the power intervals of all users in a given set on the same power axis, to facilitate comparison and manipulation through the universal function $\frac{1}{2}\ln(x)$. For simplicity, we will denote the function $\frac{1}{2}\ln(x)$ as $\mathfrak{C}$.

In Figure 2-1 the universality of $\mathfrak{C}$ and the power axis facilitate a visualization tool for interpreting manipulations in the power diagram, which we will use throughout this document.

Having thus established the equivalence between the rate and power specifications of a user in AWGN channels and the user's power interval, these two concepts will be used interchangeably in the rest of the document. Moreover, following the association of the attributes defined at the beginning of this chapter, we will often abbreviate either representation of a user's specifications simply as the user.

In the following, we develop the formulation of the AWGN multi-access achievability condition and the representation of the maximal noise threshold of sets of users in the power diagram framework to prepare for the development in the rest of the chapter.

For ease of understanding, we again start with the single-user case. By the definition of power interval, the lower boundary of a power interval is its maximal noise threshold. From

[2] In addition, notice that the $\mathsf{r}(\cdot)$ function is defined using the $\mathsf{C}(\cdot,\cdot)$ function. Therefore, $\frac{1}{2}\ln(x)$ is universal to all $\mathsf{C}(p, n_o)$ functions in the sense that for a fixed n_o, $\mathsf{C}(p, n_o) - \frac{1}{2}\ln(n_o)$ for all positive p traces out the section of $\frac{1}{2}\ln(x)$ curve for $x \geq n_o$ (after shifting the x-axis to the right for n_o units).

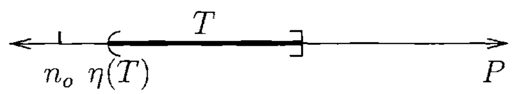


Figure 2-2: The achievability of a power interval T

the definition of the maximal noise threshold, a power interval T is therefore achievable in an AWGN single-user channel with noise variance n_o if and only if n_o is below T, as in Figure 2-2.

Definition 2.1.3. If a real number t is smaller (larger) than every element of a set A on the real line, then t is said to be *lower (higher)* than A, denoted by $t < A$ $(t > A)$.

Similarly, if two sets A, B on the real line are disjoint, and each element of A is smaller (larger) than all elements of B, A is said to be *lower* (*higher*) than B, denoted by $A < B$ $(B < A)$.

We may thus compare collections of such sets, and define the min and max functions on such collections accordingly.

With these definitions, the single-user achievability condition is

$$n_o < T \tag{2.16}$$

The multi-access achievability condition in (2.8) also has a similar expression. The following definition simplifies the notation.

Definition 2.1.4. Given a set of user power intervals $\mathcal{U} = \{T_i, i \in [1, \dots, N]\}$, define the *combined power and rate specifications* of $\mathcal{U}$ to be the sum of the power and the rate specifications of all the users in $\mathcal{U}$.

Moreover, since the combined rate and power specifications of a given set are unique, we define the *combined user* of the set $\mathcal{U}$ to be the user that has the combined rate and power specification of $\mathcal{U}$, and denote the power interval of this combined user by $\mathsf{T}(\mathcal{U})$.

In other words,

$$\mathsf{p}\left(\mathsf{T}\left(\mathcal{U}\right)\right) = \sum_{T \in \mathcal{U}} \mathsf{p}\left(T\right) \qquad \mathsf{r}\left(\mathsf{T}\left(\mathcal{U}\right)\right) = \sum_{T \in \mathcal{U}} \mathsf{r}\left(T\right) \tag{2.17}$$

To demonstrate computing the combined user, consider two users $(N = 2)$ with power intervals $(a_i, b_i], i = 1, 2$. Let their combined user power interval be $(a, b]$. Then

$$\mathsf{p}((a,b]) = \sum_{i=1}^{n} \mathsf{p}((a_i,b_i]) \iff b - a = (b_1 - a_1) + (b_2 - a_2) \tag{2.18}$$

$$\mathsf{r}((a,b]) = \sum_{i=1}^{n} \mathsf{r}((a_i,b_i]) \iff \frac{1}{2}\ln\left(\frac{b}{a}\right) = \frac{1}{2}\ln\left(\frac{b_1}{a_1}\right) + \frac{1}{2}\ln\left(\frac{b_2}{a_2}\right) \tag{2.19}$$

Multiplying both sides of (2.19) by 2 and exponentiating,

$$\frac{b}{a} = \frac{b_1}{a_1}\frac{b_2}{a_2} \tag{2.20}$$

From (2.18) and (2.20), we have

$$\begin{cases} a = \frac{(b_1-a_1)+(b_2-a_2)}{\frac{b_1}{a_1}\frac{b_2}{a_2}-1} \\ b = a + (b_1 - a_1) + (b_2 - a_2) \end{cases} \tag{2.21}$$

The combined power interval of an arbitrary set of N users may be obtained by simply extending the above formulae. Observe that only basic arithmetic operations (i.e. addition, subtraction, multiplication, and division) are involved in computing the combined user power interval from the individual power intervals in the set.

Returning to the N-user AWGN multi-access achievability, for $i \in [1, \dots, N]$, let T_i denote the power interval of user i, i.e. $\mathsf{p}(T_i) = p_i$ and $\mathsf{r}(T_i) = r_i$. For a given $\mathcal{S} \subseteq [1, \dots, N]$, let $\mathcal{V}_\mathcal{S}$ denote the subset of users with indices in $\mathcal{S}$. By definition, $\mathsf{T}(\mathcal{V}_\mathcal{S})$ is the combined user power interval of users with indices in $\mathcal{S}$, i.e.

$$\mathsf{r}(\mathcal{V}_\mathcal{S}) = \sum_{i\in\mathcal{S}} \mathsf{r}(T_i) \tag{2.22}$$

$$\mathsf{p}(\mathcal{V}_\mathcal{S}) = \sum_{i\in\mathcal{S}} \mathsf{p}(T_i) \tag{2.23}$$

By the definition of power intervals, the lower boundary of $\mathsf{T}(\mathcal{V}_\mathcal{S})$ is the maximal noise threshold of this combined user, which has the specification $\left(\sum_{i\in\mathcal{S}} \mathsf{p}(T_i), \sum_{i\in\mathcal{S}} \mathsf{r}(T_i)\right)$. By definition, the maximal noise threshold of this set of N users is $\min_{\mathcal{S}\in[1,\dots,N]} \eta(\mathsf{T}(\mathcal{V}_\mathcal{S})$ This set of users is achievable in an AWGN multi-access channel with noise variance n_o if and only if

$$n_o < \min_{\mathcal{S}\in[1,\dots,N]} \eta(\mathsf{T}(\mathcal{V}_\mathcal{S}) \tag{2.24}$$

This condition may be alternatively expressed using Definition 2.1.3 as

$$n_o < \mathsf{T}(\mathcal{V}_S), \forall S \subseteq [1, \dots, N] \tag{2.25}$$

In the rest of this chapter, we explore structures among terms on the right-hand-side of this expression, i.e. within the set

$$\{\mathsf{T}(\{T_i, i \in S\}), \forall S \subseteq [1, \dots, N]\} \tag{2.26}$$

In particular, we will show that simple checks in the power diagram framework exists for determining whether some of the terms in (2.26) may contain the others, and therefore have smaller lower boundaries. We will show that these simple checks lead to the desired simplification of the AWGN multi-access achievability condition. We start with the 2-user case.

2.2 The Two-user AWGN Multi-Access Achievability

We simplify the AWGN multi-access achievability under our new perspective using the power diagram framework in the rest of this chapter. As mentioned earlier, we address the alternate but equivalent question

> what is the *maximal noise threshold* for a given set of user specifications?

In particular, the simplification of the achievability condition results from exploring the conditions for some of the terms in (2.26) to contain the others.

As we will see, the case of two users captures the essence for the case of n users. In addition, the treatment of the two-user case eases the transition into the power diagram realm. Hence this section develops the low complexity solution to two-user achievability using the power diagram.

We start with a couple of definitions for convenience of notation.

Definition 2.2.1. The *extent* of a set of left-open-right-closed intervals $\mathcal{U}$ is the smallest left-open-right-closed interval (in length) that contains all members in $\mathcal{U}$, and is denoted $\mathsf{ext}\,(\mathcal{U})$.

Therefore, the set of power intervals $\{(a_i, b_i], i \in [1, \dots, N]\}$ has extent

$$\mathsf{ext}\,(\{(a_i, b_i], i \in [1, \dots, N]\}) = \left(\min_{i=1}^{N} a_i, \max_{i=1}^{N} b_i\right] \tag{2.27}$$

Definition 2.2.2. A set of left-open-right-closed intervals $\mathcal{U}$ is said to be *adjacent* if its members are disjoint and their union is a single left-open-right-closed interval.

For convenience, we will use $\overline{T}$ to denote the closure of a set T, and T^o to denote the interior of T. Now we state the main theorem of this section.

Theorem 2.2.1. *Let T_1, T_2 be two power intervals, and T be the combined power interval, i.e. $T = \mathsf{T}(\{T_1, T_2\})$. Then*

- *if $T_1 \cap T_2 \neq \emptyset$, the closure of $\mathsf{ext}(\{T_1, T_2\})$ is contained in the interior of T, i.e.*

$$\overline{\mathsf{ext}(\{T_1, T_2\})} \subset T^o$$

- *if T_1 and T_2 are adjacent, their combined user is equal to both their extent and their union, i.e. $T = \mathsf{ext}(\{T_1, T_2\}) = T_1 \cup T_2$.*

- *otherwise, T_1 and T_2 are separated, and $\overline{T} \subset \mathsf{ext}(\{T_1, T_2\})^o$.*

The middle piece of this three part theorem is provided by the following well known lemma.

Lemma 2.2.2.

$$\mathsf{C}(p_1, n_o) + \mathsf{C}(p_2, n_o + p_1) = \mathsf{C}(p_1 + p_2, n_o) \tag{2.28}$$

Proof.

$$\begin{aligned}\mathsf{C}(p_1, n_o) + \mathsf{C}(p_2, n_o + p_1) &= \frac{1}{2}\left(\ln\left(\frac{n_o + p_1}{n_o}\right) + \ln\left(\frac{n_o + p_1 + p_2}{n_o + p_1}\right)\right) \\ &= \frac{1}{2}\ln\left(\frac{n_o + p_1 + p_2}{n_o}\right) \\ &= \mathsf{C}(p_1 + p_2, n_o)\end{aligned} \tag{2.29}$$

□

Alternatively, this lemma may be trivially proven with the universal function, $\mathfrak{C}$. From Figure 2-1, $\mathsf{C}(p, n_o)$ is the height that $\mathfrak{C}$ spans from n_o to $n_o + p$. Therefore, this lemma only states the trivial fact that the combined height that $\mathfrak{C}$ spans from n_o to $n_o + p_1$ (height of vector $\overrightarrow{AB}$) and from $n_o + p_1$ to $n_o + p_1 + p_2$ (height of vector $\overrightarrow{BC}$) is equal to the height that $\mathfrak{C}$ spans from n_o to $n_o + p_1 + p_2$ (height of vector $\overrightarrow{AC}$), as illustrated in Figure 2-3.

Recall, also from the interpretation of the power intervals depicted in Figure 2-1, that the horizontal and vertical dimensions of a vector that is a chord on $\mathfrak{C}$ respectively specify the power

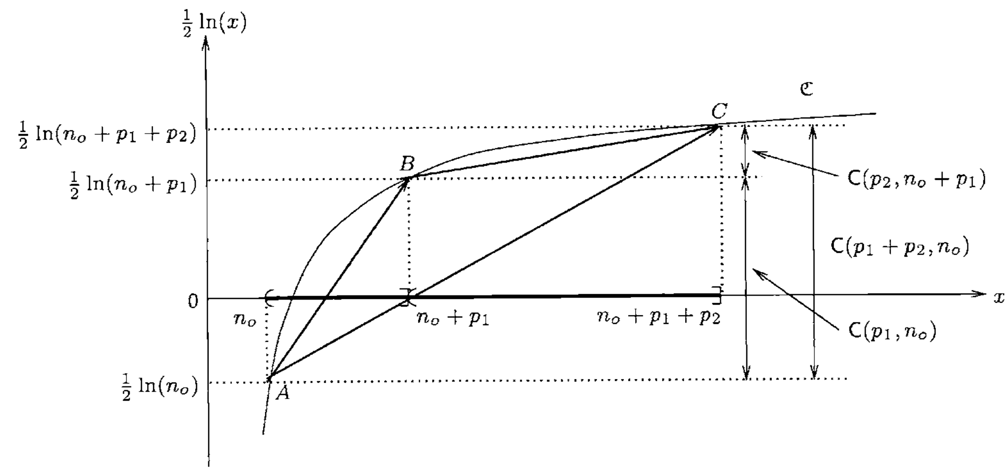


Figure 2-3: The addition of two power interval and their vector representations

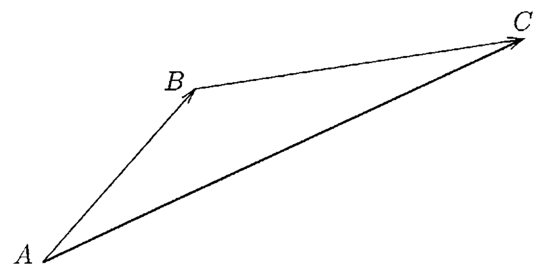


Figure 2-4: The addition of two power interval vectors

constraint and transmission rate of a user power interval which is the horizontal projection of the vector. Since $\overrightarrow{AB}+\overrightarrow{BC}=\overrightarrow{AC}$, and points A, B and C are all on $\mathfrak{C}$, we conclude that the combined specifications of $(n_o, n_o+p_1]$ and $(n_o+p_1, n_o+p_1+p_2]$ must be $(n_o, n_o+p_1+p_2]$. Observing the adjacency of $(n_o, n_o+p_1]$ and $(n_o+p_1, n_o+p_1+p_2]$, we have completed the proof of the second statement of Theorem 2.2.1.

We prove the remaining two statements of the theorem in the following.

Proof of Theorem 2.2.1: (Appendix A offers an alternate proof for the readers' indulgence.)

The proof hinges on the property of the $\mathsf{r}(\cdot)$ function.

We use the vector interpretation of power intervals as in depicted in Figure 2-1. By the definition of combined user, the vector corresponding to $T=\mathsf{T}(\{T_1,T_2\})$ must be the sum of the two vectors corresponding to T_1 and T_2. As depicted in Figure 2-4, $\overrightarrow{AB}$, $\overrightarrow{BC}$ and $\overrightarrow{AC}$ are the vectors corresponding to T_1, T_2 and T, respectively. Without the loss of generality, we have assumed that the slope of $\overrightarrow{AB}$ is greater than that of $\overrightarrow{BC}$ so that point B is above the line passing through points A and C.

Consider moving the triangle ABC so that both A and C are on the universal curve $\mathfrak{C}$. From the interpretation of Figure 2-1, the horizontal locations of A and C mark the beginning

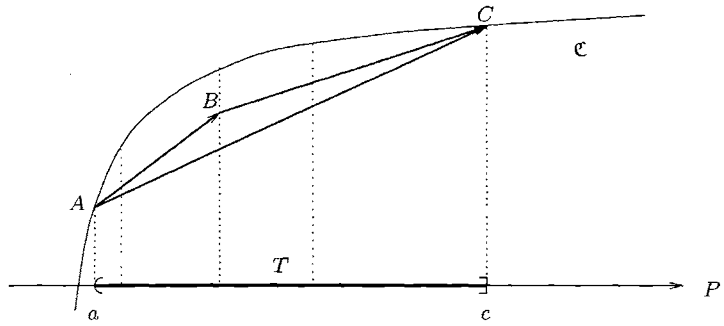


Figure 2-5: When B is below $\mathfrak{C}$

and the ending of T. Observe that there are three possible locations of point B with respect to the universal curve $\mathfrak{C}$. They are above, below, and on $\mathfrak{C}$.

The case when B is on $\mathfrak{C}$ is depicted in Figure 2-3. We have shown, in Lemma 2.2.2, that this corresponds to the case when T_1 and T_2 are adjacent. In the rest of this proof, we consider the remaining two possibilities.

Let $T = (a, c]$, and let b be the horizontal location of point B.

First, we consider the case when B is below $\mathfrak{C}$. This is depicted in Figure 2-5. We will show that in this case, T_1 and T_2 intersect and $\overline{\mathsf{ext}(\{T_1, T_2\})} \subset T^o$.

The proof for this case is a simple consequence of the fact that given two chords on $\mathfrak{C}$ with identical direction (ratio of height over length), the horizontal projection of the longer chord contains that of the shorter. This may be graphically proven. As illustrated in Figure 2-6, $\overrightarrow{A'C'}$ and $\overrightarrow{D'E'}$ are two chords of $\mathfrak{C}$ with identical direction (slope). Hence, the fact that $\overrightarrow{A'C'}$ is longer than $\overrightarrow{D'E'}$ implies that there exists points B' and B'' between A' and C' such that $\overrightarrow{D'E'} = \overrightarrow{A'B'} = \overrightarrow{B''C'}$.

Using a parallelogram, point B' is found graphically as depicted in Figures 2-6a. Let a', d', e' and c' respectively denote the horizontal location of points A', D', E' and C'. Since B' is on the chord $\overrightarrow{A'C'}$, it is below $\mathfrak{C}$. Thus B' is below E', and A' is below D'. By the monotonicity of $\mathfrak{C}$, we conclude that A' is also to the left of D', i.e. $a' \leq d'$.

Similarly, we locate point B'' graphically as in Figures 2-6a, and let b'' denote its horizontal location. Following the same reasoning, we conclude that $e' \leq c'$. This completes the proof of the fact regarding chords with identical directions.

Return to the proof of the case when B is below $\mathfrak{C}$. Since B is below $\mathfrak{C}$, we may extend the vectors $\overrightarrow{AB}$ and $\overrightarrow{BC}$ respectively into chords $\overrightarrow{AE}$ and $\overrightarrow{DE}$ on $\mathfrak{C}$. Applying the fact regarding chords with identical directions above, we conclude that the horizontal projection of the chord on $\mathfrak{C}$ with identical dimensions as $\overrightarrow{AB}$ is contained in that of $\overrightarrow{AE}$; and the horizontal projection of the chord on $\mathfrak{C}$ with identical dimensions as $\overrightarrow{BC}$ is contained in that of $\overrightarrow{DC}$. In other words,

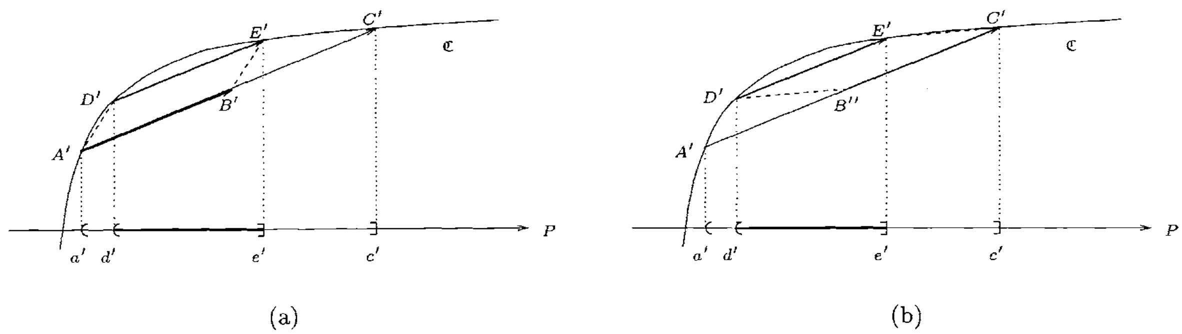


Figure 2-6: Given two chords on $\mathfrak{C}$ with identical direction, the horizontal projection of the longer chord contains that of the shorter

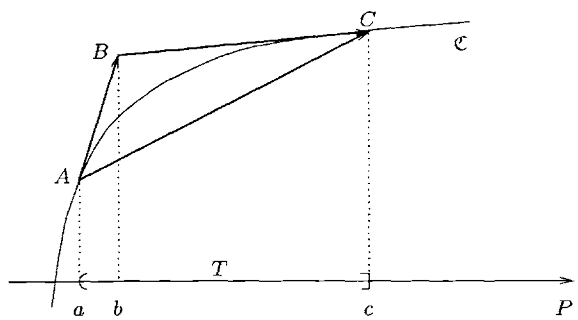


Figure 2-7: When B is above $\mathfrak{C}$

$a < T_1$ and $T_1 < c$. This completes the proof of this case.

We now consider the case when B is above $\mathfrak{C}$. This is depicted in Figure 2-7. We will show that in this case, T_1 and T_2 are separated and $\overline{T} \subset \mathsf{ext}(\{T_1, T_2\})^o$.

Notice that the vertical span of a vector starting from A and ending at the point on $\mathfrak{C}$ with horizontal location b must be less than that of $\overrightarrow{AB}$. In other words, $\mathsf{r}((a, b]) < \mathsf{r}(T_1)$. Since $b = a + \mathsf{p}(T_1)$, applying the monotonicity of $\mathsf{r}((x, x + \mathsf{p}(T_1)])$ with x, we have

$$T_1 < b \tag{2.30}$$

Similarly, notice that the vertical span of a vector starting from the point on $\mathfrak{C}$ at the horizontal location of B ending at the point C must be greater than that of $\overrightarrow{BC}$. In other words, $\mathsf{r}((b, c]) > \mathsf{r}(T_2)$. Since $b = a + \mathsf{p}(T_1)$, applying the monotonicity of $\mathsf{r}((x, x + \mathsf{p}(T_1)])$ with x, we have

$$b < \overline{T}_2 \tag{2.31}$$

Combining with (2.30), we conclude that T_1 and T_2 are separated. In addition, since (2.30) implies the lower boundary of T_1 is below a and (2.31) implies the upper boundary of T_1 is above c, we have the desired result in this case.

Since the three cases exhaust all possibilities, we have thus completed the proof of this theorem. □

Now, the consequence of this theorem.

By definition, the maximal noise threshold for two user power intervals T_1 and T_2 is

$$\min\{\eta(T_1), \eta(T_2), \eta(\mathsf{T}(\{T_1, T_2\}))\} \tag{2.32}$$

The theorem states that whether two user power intervals intersect determines which of T_1, T_2 and $\mathsf{T}(\{T_1, T_2\})$ has the least lower boundary[3]. Specifically,

1. the condition that T_1 and T_2 are intersecting or adjacent implies

$$\eta(\mathsf{T}(\{T_1, T_2\})) \leq \eta(T_1), \eta(T_2)$$

   Hence the *maximal noise threshold* in this case is $\eta(\mathsf{T}(\{T_1, T_2\}))$.

2. the condition that T_1 and T_2 are separated implies $\min\{\eta(T_1), \eta(T_2)\} \leq \eta(\mathsf{T}(\{T_1, T_2\}))$. Hence the *maximal noise threshold* in this case is $\min\{\eta(T_1), \eta(T_2)\}$.

This leads to the following algorithm for checking the two-user achievability.

Algorithm 2.2.3.

Let the two users in a two-user AWGN multi-access channel be specified by their power intervals T_1 and T_2, and the channel noise variance be n_o. Then

1. *If $T_1 \cap T_2 \neq \emptyset$ or T_1 and T_2 are adjacent, the two-user specifications are achievable in this channel if and only if $n_o < T(\{T_1, T_2\})$.*

2. *Otherwise, T_1 and T_2 are separated. The two-user specifications are achievable if and only if $n_o < \min\{T_1, T_2\}$.*

Whereas the conventional check of the two-user achievability computes capacities with logarithms, this algorithm performs simple arithmetic operations (once the power intervals of the two users are computed). Indeed, when the computations of the two users' power intervals

[3]The condition for the middle statement of Theorem 2.2.1 marks the boundary between the other two, and the corresponding case concerning the determination of the maximal noise threshold can therefore be absorbed into either of the other two cases.

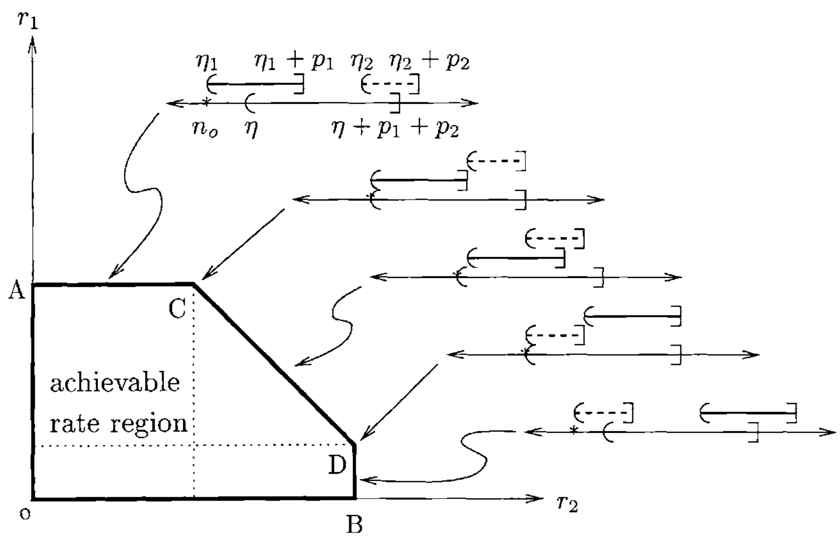


Figure 2-8: Overlapping power intervals and placement of rate tuples on 2-user AWGN multi-access achievable rate region

from their rate and power specifications are included, the actual computational requirements of the two methods differ little. However, the new algorithm demonstrates that containment relationships do exist among the terms in (2.26), and such relationships can be simply checked in the power diagram.

In the next section, we will look for simple checks of such containment relationships among the terms in (2.26) for N arbitrary users, again using the power diagram. We will show that the savings achieved in checking the existence of such conditions reduce the complexity of the achievability conditions to $O(N \ln N)$.

To help develop some intuitive understanding of Theorem 2.2.1, we place its statements in the universal function context of Figure 2-1. Observe that if the two user power intervals overlap and their maximal noise threshold is the actual channel noise variance, then neither user's rate specification may be increased without violating the achievability condition. This case corresponds to a rate tuple on the dominant face (segment CD in Figure 2-8) of the achievable rate region.

If the two power intervals do not overlap and the maximal noise threshold is the actual channel noise variance, then the rate specification of one user may be increased without violating the achievability condition. Since the other user cannot increase its rate specification without violating the achievability conditions, this case corresponds to a rate tuple on the boundary of the achievable rate region other than the dominant face. In particular, if $\eta(T_1) < \eta(T_2)$, then the rate tuple is on segment AC; otherwise, it is on segment BD. This is illustrated in Figure 2-8.

Another way to view this is that Algorithm 2.2.3 does not directly consider the achievable rate region of the actual multi-access channel as in the conventional check. Instead, it starts with the achievable rate regions of all two-user AWGN multi-access channels, and selects the one channel for which the given user specifications correspond to a rate tuple on the boundary of its achievable rate region. The achievability of the two-user specification in the given AWGN multi-access channel is determined as a simple consequence of the selection.

Before leaving this section, we generalize the third statement of Theorem 2.2.1 for later use.

Corollary 2.2.4. *Let $\mathcal{W}$ be a set of separated power intervals. Then* $\overline{\mathsf{T}(\mathcal{W})} \subset \mathsf{ext}(\mathcal{W})^o$.

Proof. Order $\mathcal{W} = \{T_i, i \in [1, \dots, K]\}$ $(K > 1)$ such that $T_i < T_{i+1}$.

By the third part of Theorem 2.2.1, there is

$$\overline{\mathsf{T}(\{T_1, T_2\})} \subset \mathsf{ext}(\{T_1, T_2\})^o \tag{2.33}$$

If $K = 2$, this completes the proof.

Otherwise, there exists a T_3. Since T_1 and T_2 are both below T_3 by assumption, $\mathsf{ext}(\{T_1, T_2\})$ is below T_3. By (2.33), $\overline{\mathsf{T}(\{T_1, T_2\})}$ must also be below T_3, i.e. $\mathsf{T}(\{T_1, T_2\})$ and T_3 must be disjoint. Now, we apply the third part of Theorem 2.2.1 a second time to $\{\mathsf{T}(\{T_1, T_2\}), T_3\}$ to obtain

$$\begin{aligned} \overline{\mathsf{T}(\{T_i, i \in [1, \dots, 3]\})} &= \overline{\mathsf{T}(\{\mathsf{T}(\{T_1, T_2\}), T_3\})} \\ &\subset \mathsf{ext}(\{\mathsf{T}(\{T_1, T_2\}), T_3\})^o \subset \mathsf{ext}(\{T_i, i \in [1, \dots, 3]\})^o \end{aligned} \tag{2.34}$$

The first equality is by definition of combined user, and the last containment is by (2.33) and the definition of extent. This completes the proof for $K = 3$. Repeating this argument, we have this lemma for arbitrary K. □

2.3 The N-user AWGN Multi-Access Achievability

In this section, we extend the results in Section 2.2 to simplify the AWGN multi-access achievability for the general case of N users. Again, we approach the AWGN multi-access achievability from the perspective resulting from the new associations of the attributes defined at the beginning of this chapter. Specifically, we consider first extracting the *maximal noise threshold* from the given set of user specifications. This is then compared to the channel noise variance to determine the achievability of the set.

As illustrated in the two-user case, the desired simplification results from properties of the set of power intervals that lead to containment relationships among terms in (2.26). In this section, we first define the *overlapping* property that plays a similar role in simplifying the N-user achievability conditions as the *intersection or adjacent* property does in the two-user case. Then we prove its consequence on simplifying the N-user achievability conditions. Finally, we present and analyze the resulting algorithm that checks N-user achievability and show it only requires $O(N \ln N)$ computations.

Consider the following definition:

Definition 2.3.1. A set of user specifications is said to have the *overlapping property*, or to be *overlapping*, if the combined user power interval of each subset is contained in the combined user power interval.

Observe that a singleton set $\mathcal{U}$ has the overlapping property. Moreover, in the case of 2-user sets, the overlapping property is equivalent to the non-separation of the two power intervals by Theorem 2.2.1.

Observe that containment of one power interval T in another S implies that the lower boundary of S is below that of T. Let $\mathcal{U}$ be an overlapping set of power intervals. Then

$$\eta(\mathsf{T}(\mathcal{U})) < \mathsf{T}(\mathcal{V}), \quad \forall \mathcal{V} \subseteq \mathcal{U} \tag{2.35}$$

Combining with (2.25), we have the following theorem.

Theorem 2.3.1. *The* maximal noise threshold *of an overlapping set $\mathcal{U}$ is the lower boundary of its combined power interval* $\mathsf{T}(\mathcal{U})$.

In other words, the achievability of an overlapping set may be checked with a single comparison, i.e. whether

$$n_o \leq \mathsf{T}(\mathcal{U}) \tag{2.36}$$

However, for the overlapping property to be useful in simplifying the N-user achievability conditions, two issues need to be resolved. First, even though the achievability of an overlapping set in a given channel may be checked with a single condition, exhaustively asserting the overlapping property of a set of N users requires $2^N - 1$ checks, which incurs the same amount of computation as exhaustively checking the N-user achievability. Therefore, for the overlapping property to be useful in simplifying the achievability conditions of overlapping sets, there need to be simpler methods of checking for the overlapping property. The second issue is that in general, a set of power intervals may not have the overlapping property. So we also need to find out how the overlapping property may help to simplify the achievability conditions for general sets.

As will be seen shortly, both issues are resolved with better understanding of the overlapping property. First, we develop an $O(N \ln N)$ algorithm that checks whether a given set of user specifications $\mathcal{U}$ has the overlapping property.

Theorem 2.3.2. *Let $\mathcal{U}$ be a set of overlapping power intervals, and $\mathcal{V} \subseteq \mathcal{U}$. Then $\mathcal{U} \setminus \mathcal{V} \cup \{\mathsf{T}(\mathcal{V})\}$ has the overlapping property.*

Proof. Let $\mathcal{W} = \mathcal{U} \setminus \mathcal{V}$ (the set difference of $\mathcal{U}$ and $\mathcal{V}$), and $\mathcal{W}_1 \subseteq \mathcal{W}$. Hence $\mathcal{W}_1 \subseteq \mathcal{U}$, and $\mathcal{W}_1 \cup \mathcal{V} \subseteq \mathcal{U}$. Since $\mathcal{U}$ is an overlapping set by definition, we have

$$\mathsf{T}(\mathcal{W}_1 \cup \mathcal{V}) \subseteq \mathsf{T}(\mathcal{U}) \tag{2.37}$$

Since $\mathsf{T}(\mathcal{W}_1 \cup \mathcal{V}) = \mathsf{T}(\mathcal{W}_1 \cup \{\mathsf{T}(\mathcal{V})\})$ and $\mathsf{T}(\mathcal{U} \setminus \mathcal{V} \cup \{\mathsf{T}(\mathcal{V})\}) = \mathsf{T}(\mathcal{U})$ by the definition of the combined user, the proof is complete. □

Applying this theorem repeatedly, we arrive at the following corollary.

Corollary 2.3.3. *Let $\mathcal{U}$ be an overlapping set, and $\{\mathcal{U}_j, j \in [1, \dots, M]\}$ be a disjoint partition of $\mathcal{U}$. Then $\{\mathsf{T}(\mathcal{U}_j), j \in [1, \dots, M]\}$ has the overlapping property.*

Theorem 2.3.4. *Let $\mathcal{U}$ be a set of power intervals, and let $\mathcal{V}$ be a subset of $\mathcal{U}$ that has the overlapping property. Then $\mathcal{U}$ has the overlapping property if and only if $\mathcal{U} \setminus \mathcal{V} \cup \mathsf{T}(\mathcal{V})$ has the overlapping property.*

Proof. If $\mathcal{U}$ has the overlapping property, then Theorem 2.3.2 shows that $\mathcal{U} \setminus \mathcal{V} \cup \mathsf{T}(\mathcal{V})$ has the overlapping property.

For the other direction, we assume that $\mathcal{U}'$, defined as $\mathcal{U} \setminus \mathcal{V} \cup \mathsf{T}(\mathcal{V})$, has the overlapping property. Let $\mathcal{W}' \subseteq \mathcal{U} \setminus \mathcal{V}$, and $\mathcal{V}' \subseteq \mathcal{V}$. We need to show that $\mathsf{T}(\mathcal{V}' \cup \mathcal{W}') \subseteq \mathsf{T}(\mathcal{U})$. Since, by definition of combined user, $\mathsf{T}(\mathcal{U}) = \mathsf{T}(\mathcal{U}' \setminus \{\mathsf{T}(\mathcal{V})\} \cup \mathcal{V}) = \mathsf{T}(\mathcal{U}')$, we may equivalently show

$$\mathsf{T}(\mathcal{V}' \cup \mathcal{W}') \subseteq \mathsf{T}(\mathcal{U}') \tag{2.38}$$

.

Since $\mathcal{W}' \subseteq \mathcal{U} \setminus \mathcal{V} \subseteq \mathcal{U}'$, by assumption, we have

$$\mathsf{T}(\mathcal{W}') \subseteq \mathsf{T}(\mathcal{U}') \tag{2.39}$$

We have (2.38) for the case when $\mathcal{V}' = \emptyset$.

Let $\mathcal{V}'$ be non-empty. Since $\mathcal{V}$ is an overlapping set by assumption, we have

$$\mathsf{T}(\mathcal{V}') \subseteq \mathsf{T}(\mathcal{V}) \tag{2.40}$$

Since $\mathsf{T}(\mathcal{V}) \in \mathcal{U}'$, the overlapping property of $\mathcal{U}'$ gives

$$\mathsf{T}(\mathcal{V}) \subseteq \mathsf{T}(\mathcal{U}') \tag{2.41}$$

. Combining with (2.40), we have

$$\mathsf{T}(\mathcal{V}') \subseteq \mathsf{T}(\mathcal{U}') \tag{2.42}$$

Suppose $\mathsf{T}(\mathcal{V}')$ and $\mathsf{T}(\mathcal{W}')$ are separated. By Theorem 2.2.1,

$$\mathsf{T}(\{\mathsf{T}(\mathcal{V}'), \mathsf{T}(\mathcal{W}')\}) \subseteq \mathsf{ext}(\{\mathsf{T}(\mathcal{V}'), \mathsf{T}(\mathcal{W}')\}) \tag{2.43}$$

By definition of combined user, $\mathsf{T}(\mathcal{V}' \cup \mathcal{W}') = \mathsf{T}(\{\mathsf{T}(\mathcal{V}'), \mathsf{T}(\mathcal{W}')\})$. Combining (2.43) with (2.39) and (2.42), we have (2.38) for this case.

Otherwise, $\{\mathsf{T}(\mathcal{V}'), \mathsf{T}(\mathcal{W}')\}$ is an overlapping set. By Theorem 2.2.1, there is

$$\mathsf{T}(\mathcal{V}') \subseteq \mathsf{T}(\{\mathsf{T}(\mathcal{V}'), \mathsf{T}(\mathcal{W}')\}) \tag{2.44}$$

Let $\mathcal{V}'' = \mathcal{V} \setminus \mathcal{V}'$. Following the argument used from (2.40) to (2.42), we have

$$\mathsf{T}(\mathcal{V}') \subseteq \mathsf{T}(\mathcal{U}') \tag{2.45}$$

Moreover, by Corollary 2.3.3, $\{\mathsf{T}(\mathcal{V}'), \mathsf{T}(\mathcal{V}'')\}$ is an overlapping set. In other words, $\mathsf{T}(\mathcal{V}')$ and $\mathsf{T}(\mathcal{V}'')$ must either intersect or be adjacent. By (2.44), so must $\mathsf{T}(\mathcal{V}'')$ and $\mathsf{T}(\{\mathsf{T}(\mathcal{V}'), \mathsf{T}(\mathcal{W}')\})$. In other words, $\{\mathsf{T}(\{\mathsf{T}(\mathcal{V}'), \mathsf{T}(\mathcal{W}')\}, \mathsf{T}(\mathcal{V}'')\}$ must be an overlapping set. By Theorem 2.2.1, we have

$$\mathsf{T}(\{\mathsf{T}(\mathcal{V}'), \mathsf{T}(\mathcal{W}')\} \subseteq \mathsf{T}(\{\mathsf{T}(\{\mathsf{T}(\mathcal{V}'), \mathsf{T}(\mathcal{W}')\}, \mathsf{T}(\mathcal{V}'')\}) \tag{2.46}$$

By the definition of the combined user, the left-hand-side is equal to $\mathsf{T}(\mathcal{V}' \cup \mathcal{W}')$, and the right-hand-side is equal to $\mathsf{T}(\mathcal{V}' \cup \mathcal{W}' \cup \mathcal{V}'') = \mathsf{T}(\{\mathsf{T}(\mathcal{V})\} \cup \mathcal{W}')$. We therefore have

$$\mathsf{T}(\mathcal{V}' \cup \mathcal{W}') \subseteq \mathsf{T}(\{\mathsf{T}(\mathcal{V})\} \cup \mathcal{W}') \tag{2.47}$$

Finally, since $\{\mathsf{T}(\mathcal{V})\} \cup \mathcal{W}'$ is a subset of $\mathcal{U}'$, by assumption, we have $\mathsf{T}(\{\mathsf{T}(\mathcal{V})\} \cup \mathcal{W}') \subseteq \mathsf{T}(\mathcal{U}')$. This completes the proof. □

One consequence of Theorem 2.3.4 is an hierarchical check of the overlapping property.

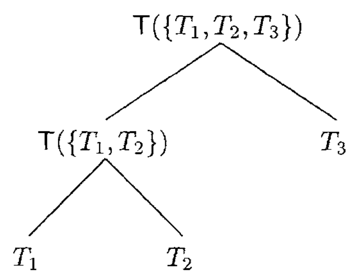


Figure 2-9: A tree of three overlapping power intervals

Since we know that a two user set has the overlapping property if the two user power intervals are non-separating from Theorem 2.2.1, we start with a set of three-user specifications $\mathcal{U} = \{T_1, T_2, T_3\}$. Let $\mathcal{V} = \{T_1, T_2\}$. If $\mathcal{V}$ has the overlapping property, then, by this theorem, $\mathcal{U}$ has the overlapping property if and only if $\{\mathsf{T}(\{T_1, T_2\}), T_3\}$ has the overlapping property.

This result may be visualized using a binary tree. Let us consider indicating the non-separation of two power intervals by assigning them to be the children of a common parent on a tree, and let such a parent node be labeled by the combined user of its two immediate children. The conclusion in the above states that if a subset of $\mathcal{U}$ containing two power intervals, i.e. $\mathcal{V}$, has the overlapping property, then $\mathcal{U}$ has the overlapping property if and only if a three-leaf binary tree can be constructed as in Figure 2-9.

Now, by Corollary 2.2.4, the extent of a set of separated power intervals contains the combined user of the set. Hence, by definition of the overlapping property, a set of power intervals having the overlapping property must contain two overlapping power intervals. In other words, if $\mathcal{U}$ has the overlapping property, we must be able to find a two-user subset $\mathcal{V}$ that has the overlapping property. Combining with the conclusion in the above, we see that $\mathcal{U}$ has the overlapping property if and only if a three-leaf binary tree can be constructed as in Figure 2-9.

A slight extension of the above argument can be used to construct trees of four power intervals with the overlapping property. To start, we use Corollary 2.2.4, as in the three user case, to show that there must exist a subset of two overlapping power intervals in the four user set. Call it $\mathcal{V}$. By Theorem 2.3.4, $\mathsf{T}(\mathcal{V})$ and the remaining two power intervals must form an overlapping set. Since the member of any set of three power intervals with the overlapping property must correspond to the leaves of Figure 2-9, a tree of four power intervals with the overlapping property must be one resulting from growing a single leaf on the tree in Figure 2-9. We therefore conclude that the members of a set of four power intervals with the overlapping property must form the leaves of one of the two types of trees in Figure 2-10.

The argument and the construction of the binary tree in the four-user case may be repeated for each additional user. We will call such a binary tree an *overlapping tree*. To emphasize, recall that in an overlapping tree,

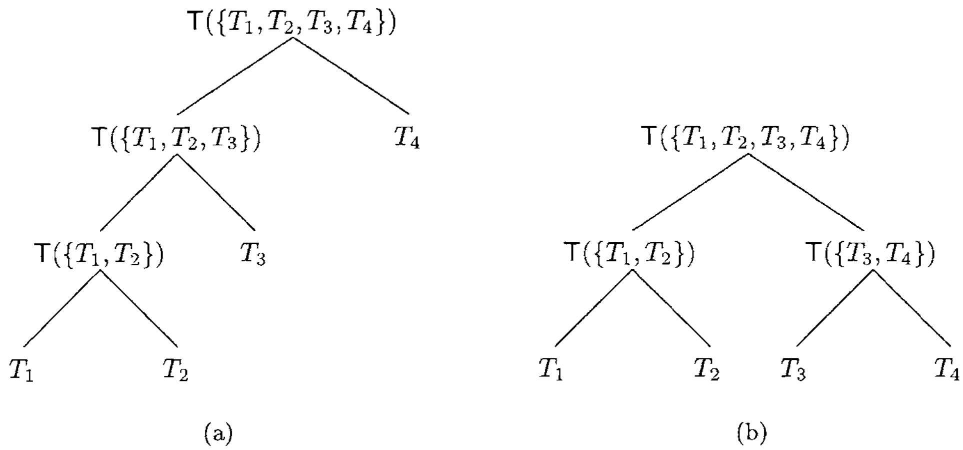


Figure 2-10: Trees of four overlapping power intervals

- the two immediate children of a node correspond to two overlapping power intervals.
- a parent node is labeled with the combined user of its two immediate children.

We have the following conclusion.

Theorem 2.3.5. *The members of a set of N power intervals with the overlapping property must constitute the leaves of an overlapping tree.*

For simplicity, we will refer to a set of power intervals with the overlapping property as an *overlapping set.*

Algorithmically, the process of constructing one such tree corresponds to successively replacing two overlapping power intervals by their combined user. We formalize this algorithm in the following.

Algorithm 2.3.6.
Let $\mathcal{U}' = \mathcal{U}$.

1. *If there exist T_1, $T_2 \in \mathcal{U}'$ such that T_1 and T_2 are overlapping (i.e. either intersecting or adjacent), then let $\mathcal{U}' = \mathcal{U}' \setminus \{T_1, T_2\} \cup \{\mathsf{T}(\{T_1, T_2\})\}$, and repeat this step*
2. *Otherwise, stop*

Note that any choice of two overlapping power intervals T_1 and T_2 may be used in each iteration of this algorithm. This is because the conclusion in Theorem 2.3.5 does not depend on which pair of overlapping power intervals is chosen to be $\mathcal{V}$ in the argument in the above. We therefore have the following theorem.

Theorem 2.3.7. *A set of user specifications $\mathcal{U}$ has the overlapping property if and only if the outcome of Algorithm 2.3.6 contains a single power interval.*

Observe that if a given set of N users has the overlapping property, $N-1$ iterations are necessary before the algorithm terminates, i.e. before a single tree is constructed. Since fewer than $N-1$ iterations are required to conclude otherwise, this algorithm must terminate with less than $N-1$ iterations. Hence the total computational burden of this algorithm depends on the operations required for finding two overlapping power intervals for each iteration.

The following realization of this algorithm starts with an ordering of the members in $\mathcal{U}$.

Algorithm 2.3.8.

Step 1: *Order members of $\mathcal{U}$ into $\mathcal{U}' = \{T_i = (a_i, b_i], i \in [1, \dots, N]\}$ such that $a_i \leq a_{i+1}$; initialize $i = 1$.*

Step 2: *If T_i is the last element in $\mathcal{U}'$, exit the algorithm. Otherwise*

Step 3: *If T_i and T_{i+1} are disjoint, i.e. $b_i < a_{i+1}$, then increase i by 1 and go to Step 2.*

Step 4: *Otherwise T_i and T_{i+1} overlap. Replace the two power intervals by their combined user $\mathsf{T}(\{T_i, T_{i+1}\})$. In particular, let $T_i = \mathsf{T}(\{T_i, T_{i+1}\})$ and update $\mathcal{U}'$ by renaming $T_j = T_{j+1}$ $\forall j \geq i+1$. If T_{i-1} exists, decrease i by 1, and go to Step 3; else go to Step 2*

Figure 2-11 presents a flowchart of this algorithm.

Observe that Algorithm 2.3.8 replaces two adjacently ordered overlapping power intervals by their combined user in Step 4. To show that this algorithm results in the same outcome as Algorithm 2.3.6, we only need to make sure that the outcome of this algorithm does not contain overlapping power intervals.

This fact may be seen through induction. First, since members of the original set of user specifications $\mathcal{U}$ is initially sorted according to the ascending order of their lower ends, if T_i and T_j overlap for any $j > i$, then T_i must overlap with every one of $\{T_k, k \in [i+1, \dots, j]\}$. In particular, this is true for $i = 1$. Therefore, when the algorithm proceeds to $i = 2$ (this may take several iterations of Step 4), T_2 and T_1 must be separated. Moreover, by Theorem 2.2.1, T_1 must be below T_2.

Suppose T_i and T_{i+1} form an overlapping set at a particular iteration of Algorithm 2.3.8, and $\{T_k, k < i\}$ is a separated set with $T_{k-1} < T_k$. Let $\mathsf{T}(\{T_i, T_{i+1}\}) = (a, b]$. By Theorem 2.2.1, a is below the lower boundary of T_i, and b is above the upper boundary of T_{i+1}. Hence a is below the lower boundary of T_{i+2} (if it exists). Therefore, after T_i and T_{i+1} are replaced by

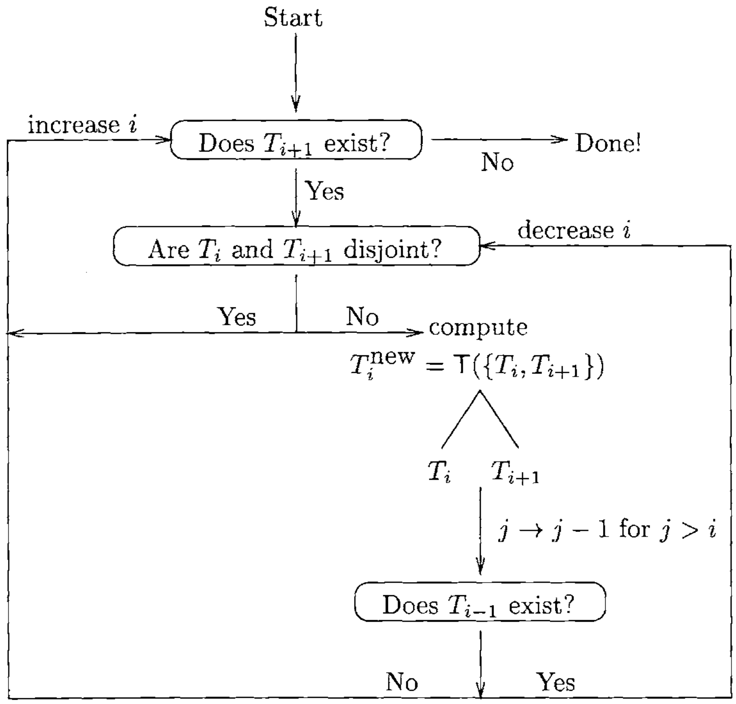


Figure 2-11: Flowchart of Algorithm 2.3.8

their combined power interval and the resulting $\mathcal{U}'$ updated, the lower boundaries of T_k for $k \geq i$ remain ordered. Similarly, b is above the lower boundary of T_{i-1} (if it exists). Hence, as in the case of $i = 1$, if $\mathsf{T}(\{T_i, T_{i+1}\})$ overlaps with T_j for any $j < i$, it must overlap with every one of $\{T_k, k \in [j, \dots, i-1]\}$.

Since a is below the lower boundary of T_i, there is the possibility that $\mathsf{T}(\{T_i, T_{i+1}\})$ may overlap with T_{i-1}. Decrementing i in Step 4 accounts for this possibility whenever such a combined user is computed. We therefore conclude that Algorithm 2.3.8 results in a set of disjoint power intervals, and is therefore one realization of Algorithm 2.3.6.

Actually, the reasoning above also shows that Algorithm 2.3.6 results in a unique set of separated power intervals, each being the root node power interval of an overlapping tree. In other words, this algorithm constructs an *overlapping forest* with members of $\mathcal{U}$ as leaves. In a moment, we will use these characteristics of the outcome of Algorithm 2.3.6 to simplify the AWGN multi-access achievability conditions for the case when the set of user specifications does not have the overlapping property.

Observe that instead of a linear array, sorting members of $\mathcal{U}$ into a linked list would reduce the computational burden of Algorithm 2.3.8. This is because no updating of the indices of members of $\mathcal{U}'$ would be necessary after replacing the two overlapping power intervals by their combined user in Step 4. Therefore, we upper-bound the computational burden of Algorithm 2.3.6 with that of Algorithm 2.3.8 when a linked list is used. Observe that once members of $\mathcal{U}$ are initially sorted, only one comparison (comparing the upper boundary of T_i with the lower boundary of T_{i+1}) is necessary to determine whether two adjacently ordered power intervals overlap. Since no more than $N-1$ computations of the combined power interval are necessary, there can be at most $2N-3$ comparisons ($N-1$ as the algorithm increments i, and $N-2$ as the algorithm decrements i). Therefore, even if the user specifications are originally given in power constraints and transmission rates, the worst case computational complexity of implementing the above algorithm is dominated by the initial sort, which is $O(N \ln N)$. This completes the simplification of the AWGN multi-access achievability conditions when the given set of user specifications has the overlapping property.

As mentioned earlier, the given set of user specifications $\mathcal{U}$ need not have the overlapping property. In such cases, the overlapping property and Algorithm 2.3.6 help to reduce the computational requirement of the AWGN multi-access achievability conditions further because Algorithm 2.3.6 partitions $\mathcal{U}$ into maximal overlapping sets.

To see this, let S_1 and S_2 be two distinct members of the set resulting from the algorithm. By the terminating condition of the algorithm, S_1 and S_2 must be separated. In particular, S_1 and S_2 are the root node power intervals of two overlapping trees with members of $\mathcal{U}$ as leaves. Let $\mathcal{U}_1$ and $\mathcal{U}_2$ be the set of leaf node power intervals of the two trees respectively. By Theorem 2.3.5, both $\mathcal{U}_1$ and $\mathcal{U}_2$ have the overlapping property. Let $\mathcal{V}_1 \subseteq \mathcal{U}_1$ and $\mathcal{V}_2 \subseteq \mathcal{U}_2$. By the definition of the overlapping property, $\mathsf{T}(\mathcal{V}_1)$ and $\mathsf{T}(\mathcal{V}_2)$ are contained in S_1 and S_2 respectively. Since $S_1 = \mathsf{T}(\mathcal{U}_1)$ and $S_2 = \mathsf{T}(\mathcal{U}_2)$ are separated, so must be $\mathsf{T}(\mathcal{V}_1)$ and $\mathsf{T}(\mathcal{V}_2)$. In other words, $\mathcal{U}_1$ and $\mathcal{U}_2$ are each maximal overlapping subsets of $\mathcal{U}$. The desired conclusion follows from the arbitrary choice of S_1 and S_2 in the set.

Moreover, since the outcome of Algorithm 2.3.6 with $\mathcal{U}_1$ or $\mathcal{U}_2$ does not depend on the choice of two overlapping power intervals during each iteration, neither does the outcome of this algorithm with $\mathcal{U}$. Accordingly, we make the following definitions.

Definition 2.3.2. Given a set of user power intervals $\mathcal{U}$. The resulting set of Algorithm 2.3.6, when its members are ordered in the ascending order of Definition 2.1.3, is called the *irreducible equivalent* of the original set, and is denoted by $\mathsf{irr}(\mathcal{U})$.

The i^{th} member power interval in $\mathsf{irr}(\mathcal{U})$ is denoted by $[\mathsf{irr}(\mathcal{U})]_i$.

By definition, $[\mathsf{irr}(\mathcal{U})]_i$ is lower than $[\mathsf{irr}(\mathcal{U})]_{i+1}$ for all feasible i. Recall that the resulting set of Algorithm 2.3.8 is so ordered.

Definition 2.3.3. The set of leaf power intervals that are descendents of $[\mathsf{irr}(\mathcal{U})]_i$ is termed the i^{th} *constructing set* of $\mathsf{irr}(\mathcal{U})$, and denoted by $\mathsf{B}_i(\mathcal{U})$.

As discussed above, each constructing set is a maximal overlapping subset of $\mathcal{U}$.

The definition of the overlapping property also provides that $\mathsf{B}_i(\mathcal{U})$ is the set of all members of $\mathcal{U}$ that are contained in $[\mathsf{irr}(\mathcal{U})]_i$. Specifically:

$$\mathsf{B}_i(\mathcal{U}) = \{T \in \mathcal{U} : T \subseteq [\mathsf{irr}(\mathcal{U})]_i\} \tag{2.48}$$

Combining these definitions and the definition of the overlapping property, we have the following.

Corollary 2.3.9. *The maximal noise threshold of* $\mathcal{U}$ *is the lower boundary of* $[\mathsf{irr}(\mathcal{U})]_1$.

Remark. This corollary is true whether the outcome of Algorithm 2.3.6 (the irreducible equivalent) is a singleton or not.

Proof. The definition of the overlapping property provides the case when $\mathsf{irr}(\mathcal{U})$ is a singleton set.

Let $M > 1$ be the number of power intervals in $\mathsf{irr}(\mathcal{U})$. Recall that $\{\mathsf{B}_i(\mathcal{U}), i \in [1, \dots, M]\}$ is a partition of $\mathcal{U}$.

Let $\mathcal{V}$ be a subset of $\mathcal{U}$. Construct $\mathcal{V}_i = \mathcal{V} \cap \mathsf{B}_i(\mathcal{U})$ for each i. Then $\{\mathcal{V}_i, i \in [1, \dots, M]\}$ is a partition of $\mathcal{V}$. The definition of the overlapping property provides that for each $i \in [1, \dots, M]$, we have

$$\mathsf{T}(\mathcal{V}_i) \subseteq [\mathsf{irr}(\mathcal{U})]_i \tag{2.49}$$

Since $\mathsf{irr}(\mathcal{U})$ is separated (as $M > 1$), $\{\mathsf{T}(\mathcal{V}_i), i \in [1, \dots, M]\}$ must also be disjoint.

By Corollary 2.2.4, we have

$$\overline{\mathsf{T}(\{\mathsf{T}(\mathcal{V}_i), i \in [1, \dots, M]\})} \subset \mathsf{ext}(\{\mathsf{T}(\mathcal{V}_i), i \in [1, \dots, M]\})^o \subseteq \mathsf{ext}(\mathsf{irr}(\mathcal{U}))^o \tag{2.50}$$

where the last containment is by (2.49). Finally, since the lower boundary of $\mathsf{ext}(\mathsf{irr}(\mathcal{U}))$ is the lower boundary of $[\mathsf{irr}\,(\mathcal{U})]_1$, the proof of this corollary is complete. □

Accordingly, we have the following algorithm to determine the achievability of a general (finite) set of user power intervals $\mathcal{U}$ in an AWGN multi-access channel with channel noise variance n_o.

Algorithm 2.3.10.

1. *Construct the irreducible equivalent of the given set of user power intervals*

2. *Compare $\eta([\mathsf{irr}(\mathcal{U})]_1)$ with n_o, and conclude that the given set of user specification is achievable in this channel if and only if $n_o \leq \eta([\mathsf{irr}(\mathcal{U})]_1)$*

The complexity of this algorithm depends on the complexity of the algorithm that constructs the irreducible equivalent. Recall, in the discussion above, that strictly fewer than $N-1$ computations of the combined power interval are necessary when $\mathcal{U}$ does not have the overlapping property. Thus the worst case computation burden of this achievability check is bounded by the case when the given set has the overlapping property, which is $O(N \ln N)$.

While it is possible to improve on the above algorithm to reduce the computational burden further, for example by observing that computing the combined user power interval for more than two users at once is simpler than successively combining two at a time, we conjecture that the worst case complexity can not be reduced below $O(N \ln N)$.

Incidentally, this algorithm is also readily adaptable for evaluating incremental achievability, e.g. answering the question whether an additional user changes the system's achievability. Notice that exhaustively checking all conditions for N-user AWGN multi-access achievability requires $2^N - 1$ checks for the additional user. In contrast, by Theorem 2.3.4, this method can compute the irreducible equivalent of the $N+1$-user set from that of the N-user set. Hence, assuming the irreducible equivalent of the N users is already computed, the additional computation burden is bounded by $O(M)$, where M is the number of elements in the irreducible equivalent of the initial set. Even in the worst case when M is equal to N – when the initial set consists of disjoint power intervals – this method still achieves a substantial saving.

Insights developed in the previous section for checking two-user achievability can be readily generalized to the N-user case. In particular, the irreducible equivalent of a set of N-user power intervals $\mathcal{U}$ designates whether the set corresponds to a rate tuple in the interior or on the boundary (on which bounding plane) of the achievable rate region of a given channel. Corollary 2.3.9 states that the maximal noise threshold is the lower boundary of $[\mathsf{irr}(\mathcal{U})]_1$. Let the lower boundary of $[\mathsf{irr}(\mathcal{U})]_1$ be η. If the noise variance of the given channel n_o is greater than η, then $\mathcal{U}$ is not achievable. If n_o is less than η, then $\mathcal{U}$ is achievable, and $\mathcal{U}$ corresponds to a rate tuple in the interior of the given channel's achievable rate region. Finally, if n_o is equal to η, then $\mathsf{B}_1(\mathcal{U})$ is an overlapping set with the lower boundary of its combined user at the channel noise variance n_o. Observe that in this last case, $\mathcal{U}$ corresponds to a rate tuple on a bounding plane of the given channel's achievable rate region. In particular, since the combined rate of users in $\mathsf{B}_1(\mathcal{U})$ may not be increased without violating achievability, the membership of $\mathsf{B}_1(\mathcal{U})$ determine on which bounding plane the rate tuple is located. Figure 2-12 illustrates membership of $\mathsf{B}_1(\mathcal{U})$ on the bounding planes of a 3-user multi-access achievable rate region.

Extending the above understanding, we see that for each $k \in [1, \ldots, M]$, the set of user

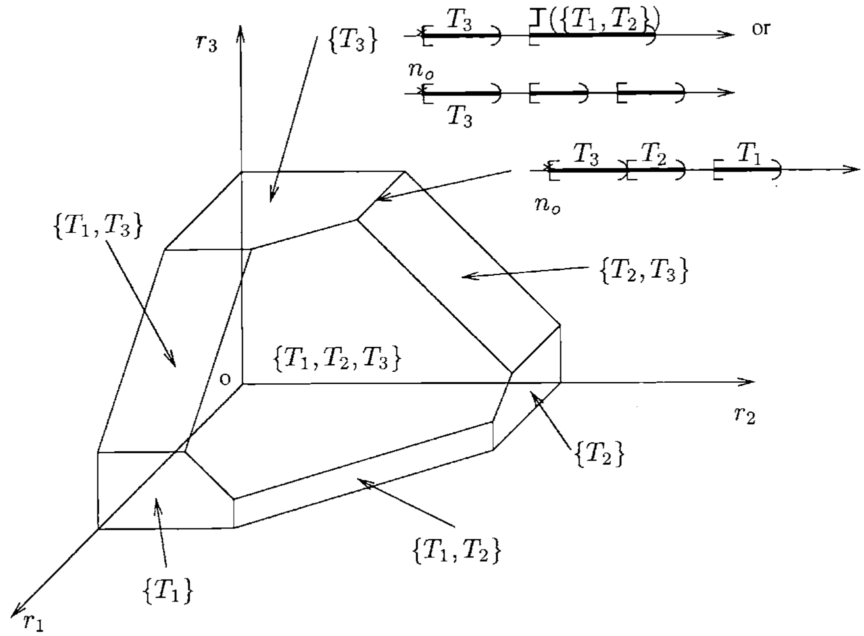


Figure 2-12: The $\mathsf{B}_1(\mathcal{U})$ on the bounding surface of three-user achievable rate region

specifications $\cup_{j \geq k} \mathsf{B}_j(\mathcal{U})$ corresponds to a point on the boundary of the multi-access achievable rate region with channel noise variance equal to the lower boundary of $\mathsf{irr}(\mathcal{U})_k$, i.e. a channel where the constructing set $\mathsf{B}_k(\mathcal{U})$ achieves its maximal combined rate possible.

In summary, constructing the irreducible equivalent selects an AWGN multi-access channel for which the given set of user specifications corresponds to a point on the boundary of its achievable rate region, just as in the 2-user case. Again, the maximal noise threshold of the given set results directly from this construction.

We have thus completed our development on simplifying the achievability of N-user AWGN multi-access channels. To recapitulate, we started by defining a new association of the user and channel attributes. This new association led to a new perspective with which characteristics of sets of user specifications such as the maximal noise threshold may be extracted independent of the channel attribute. We then developed a computational framework called the power diagram that facilitated the discovery of structures in the formulation of the characteristic. Finally, we took advantage of one such structure, called the overlapping set, and simplified the computational burden of the AWGN multi-access achievability to $O(N \ln N)$.

As a final remark, observe that Algorithm 2.3.8 may be modified to construct the irreducible equivalent from the member power interval with the greatest upper end down. We will see more constructions with the upper ends of power intervals in the next chapter.

Chapter 3

The Power Diagram and Successive Decoding in AWGN Multi-Access Channels

In the last chapter, we introduced a new perspective on AWGN multi-access channels that is based on a new association of the user and the channel attributes. Through the simplification of the AWGN multi-access achievability conditions, we showed that the advantage of this new perspective is that it allows questions regarding the set of user specifications, i.e. the maximal noise threshold of the set, to be formulated independently of the actual channel noise variance. In addition, we introduced a graphical framework called the power diagram which takes advantage of this new perspective, and presents the users and the channel separately. Finally, simplifications of the AWGN multi-access achievability condition were developed through graphical manipulations in this framework.

In the next three chapters, we approach decoding aspects of the AWGN multi-access channel from this new perspective. Specifically, the technique of *successive decoding* is treated in conjunction with *power-reduction*, *user-splitting*, and *time-sharing* techniques, each in a single chapter. These studies lead to the conclusion that the computational burden of decoding in the AWGN multi-access channel may be reduced to linear growth with the number of users in the channel. This represents a significant saving compared to the brute-force approach to joint decoding, for which the computational burden grows exponentially with the number of users. More importantly, when combined with the reduction of the achievability conditions in Chapter 2, these results lead to the conclusion that the AWGN multi-access channel is, at least in theory, no more complex than the aggregate of multiple users (or multiple AWGN single-user channels) with an ordering.

In the next section, we lay the general background and foundation for the studies which span the next three chapters. Starting with Section 3.3, we turn to study the simple successive decoding technique (to be defined) and combining power-reduction and successive decoding techniques.

3.1 Successive Decoding

The technique of successive decoding may be understood from the chain rule of mutual information (c.f. [7], page 22, (2.2.29)). This may be illustrated with a 2-user multi-access channel. Let X_1 and X_2 denote the channel inputs of users 1 and 2, and Y denote the channel output. The multi-access coding theorem dictates that the two-user rate tuple (r_1, r_2) is in the achievable rate region of this channel if and only if

$$r_1 \leq \mathsf{I}(Y; X_1|X_2) \tag{3.1}$$

$$r_2 \leq \mathsf{I}(Y; X_2|X_1) \tag{3.2}$$

$$r_1 + r_2 \leq \mathsf{I}(Y; X_1, X_2) \tag{3.3}$$

With the chain rule of mutual information, the right side of (3.3) has the following equivalence

$$\mathsf{I}(Y; X_1, X_2) = \mathsf{I}(Y; X_1) + \mathsf{I}(Y; X_2|X_1) = \mathsf{I}(Y; X_2) + \mathsf{I}(Y; X_1|X_2) \tag{3.4}$$

Consider the first equivalence: since additional conditioning does not increase entropy, there is

$$\mathsf{H}(X_1|Y) \geq \mathsf{H}(X_1|Y, X_2) \tag{3.5}$$

Hence

$$\begin{aligned} \mathsf{I}(Y; X_1) &= \mathsf{H}(X_1) - \mathsf{H}(X_1|Y) = \mathsf{H}(X_1|X_2) - \mathsf{H}(X_1|Y) \\ &\leq \mathsf{H}(X_1|X_2) - \mathsf{H}(X_1|Y\,X_2) = \mathsf{I}(Y; X_1|X_2) \end{aligned} \tag{3.6}$$

The second equality results from the (assumed) independence of the user inputs X_1 and X_2. Hence, the rate tuple $(\mathsf{I}(Y; X_1), \mathsf{I}(Y; X_2|X_1))$ satisfies all three conditions, and is therefore in the achievable rate region of this channel. By definition of the achievable rate region, for given fixed distributions of X_1 and X_2, reliable communication can be achieved at any rate tuple

(r_1, r_2) satisfying

$$r_1 < \mathsf{I}(Y; X_1) \tag{3.7}$$

$$r_2 < \mathsf{I}(Y; X_2|X_1) \tag{3.8}$$

From the coding theorem (c.f. [7]), (3.7) implies that, for the given distributions of X_1 and X_2, user 1's message can be decoded directly from Y and independently from user 2's transmission to arbitrarily small probability of error as the block length of the code grows sufficiently large. Additionally, assuming user 1's message is completely known, (3.7) implies that user 2's message can also be decoded to arbitrary small probability of error, again using transmission codes of sufficiently long block length.

Consider decoding these two users' transmission as in the following:

Step 1. decode user 1's message from Y directly

Step 2. decode user 2's message assuming the decoded message of user 1 is the *true* message sent

To bound the total decoding error using this decoding procedure, we consider the probability of error on the joint message of the two users. In particular, we will say that a decoding error occurs whenever any error occurs in decoding either user's messages. Let this quantity be denoted by P_e. Observe that, following this decoding procedure, the *no-error* event occurs when user 1's message is decoded correctly in Step 1 and user 2's message is decoded correctly in Step 2. Applying the union bound, we have

$$P_e < \epsilon_1 + \epsilon_2 \tag{3.9}$$

Since both ϵ_1 and ϵ_2 can be made arbitrarily close to 0, the same must be true for P_e. In other words, this two-step decoding procedure achieves reliable communication for this case. This procedure of decoding independent transmissions one at a time as if each user is transmitting in a single-user channel may be generalized to the N user multi-access channel case for arbitrary $N > 2$, and has been termed *successive decoding*, or *single-user decoding*.

Definition 3.1.1. Let N be a positive integer. Denote the set of all permutations of integers $[1, \ldots, N]$ by $\mathbf{\Pi}(N)$.

Specifically, for a given decoding order $\mu \in \mathbf{\Pi}(N)$, the technique of *successive decoding* decodes users $\mu(N)$ through $\mu(1)$ one at a time following the (inverse) ordering μ, each time taking into account only those user transmissions which are already decoded. Note the decoding order for the two-user example in the above is $\{1, 2\}$.

Accordingly, we will say that a rate tuple is *successively decodable* in a multi-user channel if there exists an ordering of the set of user transmissions according to which the rate tuple can be achieved in this channel using the the technique of successive decoding. Observe that by definition, a successively decodable rate tuple must be achievable. The set of all successively decodable rate tuples in a channel is called the *successively decodable rate region* of the channel.

Observe that a total of N decoding stages are necessary for a set of N users. The total computational requirement of successive decoding is equal to the sum of the computational requirement of each decoding step. In each decoding step, only one user's transmission is decoded as if in a single-user channel. To decode this user's transmission optimally, the decoder must take into account the remaining undecoded user transmissions as part of the mechanism that corrupts this user's transmission. This is accomplished through computing the transition probability from the channel input of this user to the channel output conditioned on the previous decoding results.

To illustrate this computation, consider a discrete-time memoryless multi-access channel with N users with channel transition probability $P_{Y|X_1,\ldots,X_N}$, where Y denotes the channel output, and X_i denotes user i's channel input. In addition, let Θ_i denote the set of channel inputs that user i can generate (X_i can take on), and Q_i denote the probability distribution with which X_i is chosen from Θ_i independently of the other user's messages.

Without the loss of generality, consider successively decoding this set of users with decoding order $\{1, \ldots, N\}$. In other words, user i's message is decoded assuming only the messages of users 1 through $i-1$ are known. For convenience, let $\mathbf{X}(i)$ denote the channel inputs of the set of already decoded users, i.e.

$$\mathbf{X}(i) = (X_1, \ldots, X_{i-1}) \tag{3.10}$$

and let $\hat{\mathbf{x}}(i) = (\hat{x}_1, \ldots, \hat{x}_{i-1})$ denote the set of decoded messages prior to decoding user i, where, for $j \in [1, \ldots, i-1]$, $\hat{x}_j$ is the decoded message of user j. From the above discussions, optimal decoding of user i's message requires the transition probability from user i's channel input X_i to Y conditioning on the known messages $\hat{\mathbf{x}}(i)$. Using Bayes' rule, we have the following expression of the transition probability

$$P_{Y|X_i,\mathbf{X}(i)}(y_o|\hat{\mathbf{x}}, x_i) = \sum_{x_j \in \Theta_j, j \in [i+1,\ldots,N]} P_{Y|X_1,\ldots,X_N}(y|\hat{\mathbf{x}}(i), x_i, \ldots, x_N)\, Q_{i+1}(x_{i+1}) \cdots Q_N(x_N) \tag{3.11}$$

For simplicity, assume, during each unit of time, each user can choose one of K possible channel inputs, and the channel outputs of K symbols. There may be multiple approaches to obtaining the transition probability in (3.11) for each successive decoding step.

A brute force approach is to compute directly using (3.11) for each decoding step. In this case, observe that for the i_{th} successive decoding step, the sum in the right-hand-side of (3.11) is over the joint channel inputs of users $i+1$ through N (which consists of K^{N-i} terms) for the observed channel output y_o and each possible channel input of user i in Θ_i. Since the operand in the sum is a product of $N-i+1$ terms, we conclude that computing the transition probability required for the i_{th} successive decoding step incurs $K^{N-i+1}(N-i)$ computations. Observe that computing the desired transition probability for the first decoding step requires $K^N(N-1)$ computations, which is on the same order as that of joint decoding.

Alternatively, one may consider computing all such transition probabilities *a priori* and store them in memory. Observe that, for the i_{th} successive decoding step, since neither the channel output nor the decoded messages of users 1 through $i-1$ is known, K^{i+1} memory entries are required. Since the entire $P_{Y|X_i,i\in[1,\dots,N]}$ needs to be stored for the last decoding step, we conclude that a total of $\sum_{j=1}^{N-1} K^{j+1} = O(K^{N+1})$ memory entries are required in addition to $P_{Y|X_i,i\in[1,\dots,N]}$ using this approach. Observe that the total memory requirement is greater than the computational requirement of joint decoding.

Now, observing that the computational burden for the desired transition probability decreases with i while the memory requirement for it increases with i, one may consider combining the two approaches in the above. However, the point of this exercise is to show that for general multi-access channels, the technique of successive decoding does not necessarily result in significant savings compared to joint decoding.

Also note, from the above analysis, that the reason why successive decoding does not necessarily result in significant savings for general multi-access channels is the large computation burden required to obtain the necessary transition probability during each decoding step. So, if there exists some means of obtaining these transition probabilities simply, then a substantial savings may be achieved as each decoding step requires no more computations than decoding in a single-user channel. The AWGN multi-access channels offers one such theoretically interesting case when the users are restricted to using the *Gaussian ensemble of codes*, as discussed below.

Recall that in an AWGN single-user channel with average power constraints on the user's transmission, the maximal mutual information between the user transmission and the channel output is achieved only when the channel input of the user is independent and identically distributed (IID) Gaussian [13]. Subsequently, it was shown that, in an AWGN single-user channel, when the transmission code book is chosen from the Gaussian ensemble of codes–

codewords with each channel input chosen IID according the 0-mean Gaussian distribution with variance equal to the user's power constraint – the average probability of decoding error over the entire ensemble is arbitrarily small for any transmission rate below the channel capacity [15].

In an AWGN multi-access channel with average power constraints on the users, it has been shown that the maximal combined mutual information of all users is achieved only when the channel input of each user is IID Gaussian (c.f. [8]). In 1985, Gallager [8] adapted the Gaussian ensemble of codewords for proving the coding theorem in AWGN multi-access channels by modifying the rules of choosing the codewords using the combined power constraints of all subsets of the users. He showed that when each user chooses entries in its transmission code book independently from this modified Gaussian ensemble of codewords, the average probability of any decoding error over this ensemble is arbitrarily small for any rate tuple in the channel's achievable rate region. Now, when a user transmits using such a codebook, its transmission appears, on average over the modified ensemble, to be a white Gaussian process that is independent of the other users' transmissions. Observe that by the conclusion on the distributions of users' transmissions required to achieve the maximal combined mutual information in this channel mentioned in the above, this must be the case. For simplicity, we will refer to transmission using such codebooks as using the *Gaussian ensemble of codes.*

Consider successive decoding in a 2-user AWGN multi-access channels when users each use a the Gaussian ensemble of codes. Let X_1 and X_2 be the channel inputs of users 1 and 2, Z the additive white Gaussian noise in the channel, and Y the channel output. We have $Y = X_1 + X_2 + Z$. Let n_o denote the variance of Z, and let p_1, p_2 respectively denote the power specifications of users 1 and 2. Without the loss of generality, consider successively decoding these two users with order $\{1, 2\}$.

Assuming both users choose their messages independently, since both users use Gaussian ensemble of codes, from the discussion above, X_1 and X_2 appear to be independent white Gaussian sequences (averaging over the modified Gaussian ensemble of codewords) with intensities p_1 and p_2 respectively. Therefore, $Y = X_1 + X_2 + Z$ and $X_1 + Z$ also appear to be white Gaussian processes with intensities $p_1 + p_2 + n_o$ and $p_1 + n_o$, respectively. Hence, user 1's transmission is decoded in the presence of additive white Gaussian interference (AWGI) as if in an AWGN single-user channel with noise variance $n_o + p_1$. The conclusion is that users' messages are decoded in the presence of AWGI in both successive decoding steps. Since the distributions of these AWGI only differ in their variance, we see that such a setup would reduce the computational requirement for computing the channel transition probabilities to a minimium. In particular, observe that successively decoding this set requires a total computational burden that is comparable to decoding two AWGN single-user channels.

These observations can be easily generalized to arbitrary N-user AWGN multi-access channels to conclude that, when users are restricted to using the Gaussian ensemble of codes, the total computational burden required for successive decoding is comparable to decoding N AWGN single-user channels. This represents a significant saving compared to joint decoding, which requires $O(2^N)$ computations.

For simplicity, we will refer to successive decoding in AWGN multi-access channels when users are restricted to using the Gaussian ensemble of codes as *simple successive decoding.* Accordingly, we define the following:

Definition 3.1.2. A rate tuple is said to be *SSD with* μ, in an AWGN multi-access channel if users' transmission can be decoded with simple successive decoding and this ordering in the given channel.

A rate tuple is said to be *SSD* in an AWGN multi-access channel if there exists one ordering μ such that this rate tuple is SSD with μ in this channel.

The set of all rate tuples that are SSD in a channel is said to be the *SSD rate region* of the channel.

Since averaging over the modified Gaussian ensemble of codewords gives the transmission of users the appearance of independent white Gaussian process, no single set of codewords is known to generate such appearances. However, despite this limitation, analysis with simple successive decoding remains meaningful, at least in the following two areas:

1. It provides a possibility of simplifying the decoding aspect of the AWGN multi-access channel, at least, in theory. As we will see in the two succeeding chapters, combining simple successive decoding with either user-splitting or time-sharing techniques help to simplify AWGN multi-access decoding to be comparable to decoding $2N-1$ AWGN single-user channels.

2. Since AWGN is the worst kind of noise in the sense that it minimizes the mutual information between channel inputs and output for fixed noise power, analysis with simple successive decoding helps to establish bounds on each successive decoding step. As bounds, they provide guidelines for further studies on successive decoding, and for experiments with successive decoding in real systems.

In the rest of this chapter, we approach simple successive decoding and combining simple successive decoding with the power-reduction technique from our new perspective on the AWGN multi-access channel.

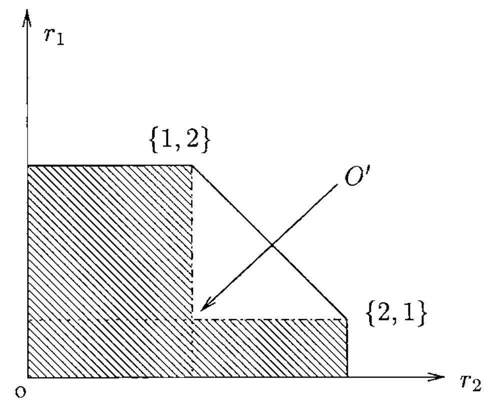


Figure 3-1: The SSD rate region in the achievable rate region of a two-user AWGN multi-access channel

Both the simple successive decoding technique and the SSD rate region in AWGN multi-access channels have been long known. We use the two-user case discussed in the above to illustrated the SSD rate region. By definition of simple successive decoding, users must use the Gaussian ensemble of codes. From the above discussions, we know that using the Gaussian ensemble of codes gives each user's transmission the appearance of IID Gaussian processes with variance equals to the user's power constraints, and the channel interference when decoding user 1 first is AWGI with variance $n_o + p_2$. Hence, the SSD rate region of this two-user AWGN multi-access channel when user 1 is decoded first must consists of rate tuples (r_1, r_2) satisfying

$$r_1 \leq \mathsf{I}(Y; X_1) = \mathsf{I}(X_1 + X_2 + Z; X_1) = \frac{1}{2} \ln \left(\frac{p_1 + p_2 + n_o}{p_2 + n_o} \right) \tag{3.12}$$

$$r_2 \leq \mathsf{I}(Y; X_2 | X_1) = \mathsf{I}(X_1 + X_2 + Z; X_2 | X_1) = \mathsf{I}(X_2 + Z; X_2) = \frac{1}{2} \ln \left(\frac{p_2 + n_o}{n_o} \right) \tag{3.13}$$

Observe that the rate tuple $\left(\frac{1}{2} \ln \left(\frac{p_1+p_2+n_o}{p_2+n_o} \right), \frac{1}{2} \ln \left(\frac{p_2+n_o}{n_o} \right) \right)$ is a corner point on the dominant face of this channel's achievable rate region, where the transmission rate of user 2 reaches a maximum. We may follow a similar development for the other decoding order $\{1, 2\}$, and reach the conclusion that the SSD rate region for this two-user case consists of all rate tuples dominated by at least one of two corner points. We illustrate this rate region in Figure 3-1. Generalizing to arbitrary N users, we conclude that the SSD rate region consists of all rate tuples dominated by at least one of the corner points on the dominant face of the AWGN multi-access achievable rate region. Figure 3-2 illustrates the SSD rate region for a three-user AWGN multi-access channel along with the the decoding orders of the corner points.

Observe that the SSD rate region is a strict subset of the AWGN multi-access achievable rate region. However, as we will see, this technique requires about half of the number of successive decoding steps of the combination techniques to be discussed in later chapters. From this point

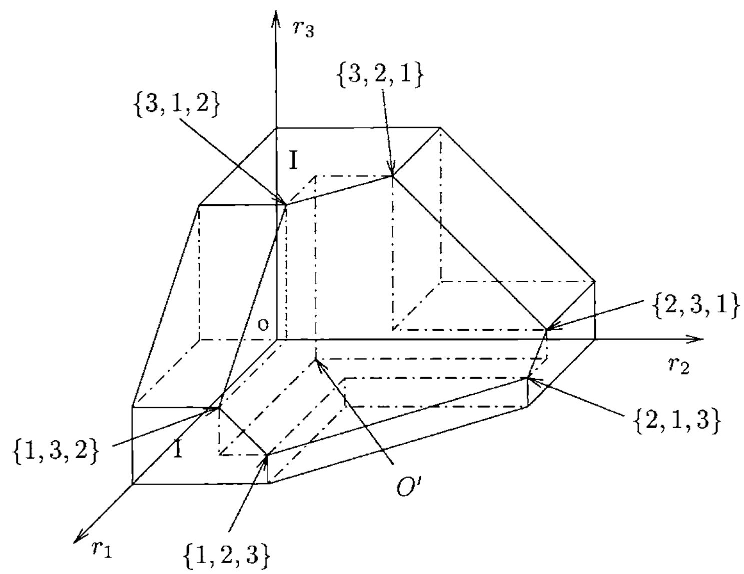


Figure 3-2: The SSD rate region, Region I, in the achievable rate region of a three-user AWGN multi-access channel

of view, when applicable, simple successive decoding is more attractive.

Starting from Section 3.3, we will propose combining simple successive decoding and power-reduction. While this combination can be used to achieve rate tuples outside the SSD rate region, it is not capable of achieving all rate tuples in the AWGN achievable rate region. We will see that this combination technique is also attractive for reasons similar to those for simple successive decoding.

The rest of this chapter is divided into two parts: first, we briefly consider simple successive decoding from our new perspective and using the power diagram framework in Section 3.2, and develop an $O(N \ln N)$ complexity algorithm that tests whether a given rate tuple is in the SSD region of a given AWGN multi-access channel. The second part starts from Section 3.3, in which we consider combining simple successive decoding and power-reduction. We develop $O(N \ln N)$ complexity algorithms that test whether a given rate tuple can be achieved with this combination of techniques in a given AWGN multi-access channel.

As we will see, the testing algorithm development in the first part employs a similar construction to those in the second part. As the former algorithm is much simpler than the latter (because of the simplicity in the techniques considered in the first part), it serves as a good entry for understanding the latter.

3.2 Simple Successive Decoding

In this section, we approach simple successive decoding in AWGN multi-access channels from the new perspective developed in Chapter 2. Recall that the new perspective separates the user and channel attributes, and allows questions concerning characteristics of set of user specifications be posed and addressed independently of the actual channel noise variance. Here, we consider the question

> what is the maximal noise threshold for a set of user specifications to be SSD?

Using the power diagram framework, we develop an algorithm that computes this maximal noise threshold from the set of N-user specifications directly requiring $O(N \ln N)$ computations. As a result, whether the rate tuple of the given set of user specifications is SSD in the given channel is determined by a simple comparison between this maximal noise threshold and the noise variance of the given channel.

The rest of this section is divided into two parts. First, we formulate the problem of *maximal noise threshold for a set of N user specifications to allow simple successive decoding* from the new perspective and using the power diagram framework. Then, we develop an $O(N \ln N)$ algorithm to compute this quantity.

3.2.1 The maximal noise threshold for simple successive decoding

Now, we develop a formulation of maximal noise threshold for a set of N user specifications to be simple successive decoding.

We start with the two-user case. For $i \in \{1, 2\}$, let T_i be the power interval with specifications (p_i, r_i). From discussions in the previous section, the rate tuple (r_1, r_2) is SSD in the given channel if and only if conditions (3.12) and (3.13) are satisfied. Consider the following equivalent to these conditions:

$$n_o + p_2 < T_1 \tag{3.14}$$

$$n_o < T_2 \tag{3.15}$$

Whereas (3.15) is given by the achievability condition of the two users, (3.14) can be interpreted as requiring the gap between channel noise variance n_o and the lower boundary of the first decoded user (T_1) be larger than the power specification (length) of T_2. This is illustrated in Figure 3-3.

Recall, from discussions of AWGN multi-access achievability in Chapter 2, a power interval with any of its portion to the left of the channel noise variance is not achievable. This is reflected

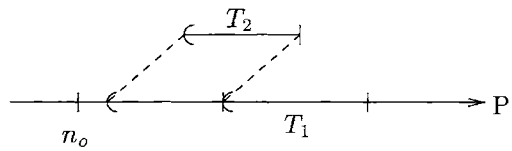


Figure 3-3: The gap between the lower boundary of T_1 and n_o must be larger than the power constraint of T_2 for user 1 to be SSD before User 2

in (3.15). Extending this understanding to (3.14), we conclude that T_1 must be to the right of $n_o + p_2$ for user 1's message to be achievable in the presence of AWGI consisting of the channel noise and user 2's transmission,

We may combine (3.14) and (3.15), and conclude that the two users are SSD with decoding order $\{1, 2\}$ if and only if

$$n_o \leq \min\{\eta(T_1) - p_2, \eta(T_2)\} \tag{3.16}$$

In other words, the maximal noise threshold for $\{T_1, T_2\}$ to be SSD with $\{2, 1\}$ is $\min\{\eta(T_1) - p_2, \eta(T_2)\}$.

Switching the roles of T_1 and T_2, we conclude that the maximal noise threshold for $\{T_1, T_2\}$ to be SSD with $\{1, 2\}$ is $\min\{\eta(T_2) - p_1, \eta(T_1)\}$. Finally, since the two users are SSD if they are SSD with either decoding order, the maximal noise threshold for $\{T_1, T_2\}$ to be SSD is

$$\max\left\{\min\left\{\eta\left(T_1\right) - p_2, \eta\left(T_2\right)\right\}, \min\left\{\eta\left(T_2\right) - p_1, \eta\left(T_1\right)\right\}\right\}$$

Or,

$$\max_{\mu \in \mathbf{\Pi}(2)} \min\left\{\eta\left(T_{\mu(1)}\right) - p_{\mu(2)}, \eta\left(T_{\mu(2)}\right)\right\} \tag{3.17}$$

Observe that the right-hand side of this condition is solely dependent on the specifications of the two users.

We now generalize this formulation to N-user AWGN multi-access channels.

For $i \in [1, \ldots, N]$, let X_i denote the channel input of user i. Let Z and Y denote the additive white Gaussian noise in the channel and the channel output respectively. We have

$$Y = \sum_{i=1}^{N} X_i + Z \tag{3.18}$$

.

Let user i be specified by power interval T_i for $i \in [1, \ldots, N]$, and n_o be the variance of the channel noise Z. Without the loss of generality, consider decoding this set of users with simple successive decoding and decoding order $\{1, \dots, N\}$. Specifically, for each $i \in [1, \dots, N]$, let user i's transmission be decoded only with the decoding results of users 1 through $i-1$, and in the presence of the channel noise and the combined interference from transmissions of users $i+1$ through N. Following the same argument used in the two-user case, we conclude that user i's transmission rate can be reliably decoded at this simple successive decoding step if and only if

$$n_o + \sum_{j=i+1}^{N} \mathsf{p}(T_j) < T_i \tag{3.19}$$

One may interpret this condition as requiring

> the gap between the channel noise variance and the lower boundary of T_i to be greater than the combined power constraint of the remaining (undecoded) users

To derive the desired maximal noise threshold quantity here, consider re-arranging the two sides of the condition to obtain

$$n_o < \eta(T_i) - \sum_{j=i+1}^{N} \mathsf{p}(T_j) \tag{3.20}$$

Note that this condition states that

> if the lower boundary of T_i is moved to the left (reduced) by an amount equal to the the combined power constraint of the remaining (undecoded) users, it should still be greater than the channel noise variance

Combining such conditions for each user, we conclude that this set of user specifications is SSD in the given channel with the given decoding order if and only if

$$n_o \leq \min_{i \in [1, \dots, N]} \eta(T_i) - \sum_{j=i+1}^{N} \mathsf{p}(T_j) \tag{3.21}$$

We have the following:

Definition 3.2.1. Given a set of user specifications $\mathcal{U} = \{T_i, i \in [1, \dots, N]\}$ and a decoding order $\mu = \{\mu(1), \dots, \mu(N)\}$, the *maximal noise threshold for $\mathcal{U}$ to be SSD with μ* is defined to be

$$\min_{i \in [1, \dots, N]} \eta\left(T_{\mu(i)}\right) - \sum_{j=i+1}^{N} \mathsf{p}\left(T_{\mu(j)}\right) \tag{3.22}$$

The above arguments gives the following theorem.

Theorem 3.2.1. *A set of user specifications is SSD with decoding order μ in an AWGN multi-access channel if and only if the channel noise variance is below the maximal noise threshold for this set to be SSD using μ.*

Since a set of user specifications is SSD in a channel if it is SSD with at least one decoding order, we have the following.

Definition 3.2.2. The *maximal noise threshold for a set of user specifications* $\{T_i, i \in [1, \cdots, N]\}$ *to be SSD* is defined to be

$$\max_{\mu \in \mathbf{\Pi}(N)} \min_{i \in [1, \cdots, N]} \eta\left(T_{\mu(i)}\right) - \sum_{j=i+1}^{N} \mathsf{p}\left(T_{\mu(j)}\right) \tag{3.23}$$

The discussion above provides the following theorem.

Theorem 3.2.2. *A set of user specifications is SSD in an AWGN multi-access channel if and only if the channel noise variance is below the maximal noise threshold for the set to be SSD.*

Since there are $N!$ distinct decoding orders for a set of N users, a brute force computation of this maximal noise threshold quantity using (3.23) requires a multitude of $N!$ computations. In the next subsection, we develop an algorithm to compute it through a construction requiring $O(N \ln N)$ computations.

3.2.2 Computing the maximal noise threshold for simple successive decoding with the p-maximal allocation set

We now develop an $O(N \ln N)$ algorithm for computing the maximal noise threshold for a given set of N user specifications $\mathcal{U}$ to be SSD.

From its formulation in (3.23), note that the maximal noise threshold for $\mathcal{U}$ to be SSD is given by the maximum of the maximal noise threshold for $\mathcal{U}$ to be SSD among all possible decoding orders. Now, suppose μ^* is the ordering with which the maximum is attained. Then the maximal noise threshold for $\mathcal{U}$ to be SSD is the maximal noise threshold for $\mathcal{U}$ to be SSD with μ^*. In other words, μ^* can be used to simple successively decode $\mathcal{U}$ in every channel in which $\mathcal{U}$ is SSD. In this sense, μ^* is an *optimal ordering.*

We take advantage of the above observation, and achieve the desired simplification of the formulation of this maximal noise threshold in three steps. First, we investigate the simplification of the maximal noise threshold for simple successive decoding with a particular decoding order. Then, we illustrate that this optimal ordering is determined by the ordering of power intervals in $\mathcal{U}$ by their upper boundaries using the two-user case. Finally, we define a construction for $\mathcal{U}$, called the p-*maximal allocation set*, that combines the results achieved in the first two steps. We show that the desired maximal noise threshold here is given by an appropriate property of the p-maximal allocation set.

We start by investigating the simplification of the maximal noise threshold for simple successive decoding with a particular decoding order. Without the loss of generality, we consider the ordering $\{1, \ldots, N\}$. From (3.22), the maximal noise threshold quantity we are concerned here is

$$\min_{i \in [1, \ldots, N]} \eta(T_i) - \sum_{j=i+1}^{N} \mathsf{p}(T_j) \tag{3.24}$$

For simplicity, let

$$\eta_i = \eta(T_i) - \sum_{j=i+1}^{N} \mathsf{p}(T_j), \quad i \in [1, \ldots, N] \tag{3.25}$$

Hence (3.24) is equivalent to $\min_{i \in [1, \ldots, N]} \eta_i$.

Observe that

$$\eta_i - \eta_{i+1} = \eta(T_i) - \mathsf{p}(T_{i+1}) - \eta(T_{i+1}) = \eta(T_i) - (\eta(T_{i+1}) + \mathsf{p}(T_{i+1})) \tag{3.26}$$

In particular, comparison of adjacent terms in the minimization, η_i and η_{i+1}, is determined by the comparison of the lower boundary of T_i and the upper boundary of T_{i+1}. In addition,

$$\eta_i - \eta_{i+l} = \eta(T_i) - \sum_{j=i+1}^{i+l} \mathsf{p}(T_j) - \eta(T_{i+l}) = \left(\eta(T_i) - \sum_{j=i+1}^{i+l-1} \mathsf{p}(T_j) \right) - (\eta(T_{i+l}) + \mathsf{p}(T_{i+l})) \tag{3.27}$$

In other words, comparison of terms in the minimization, η_i and η_{i+l}, is determined by the comparison of the lower boundary of T_i reduced by the combined power specifications of users $i+1$ through $i+l-1$ and the upper boundary of T_{i+l}. We may therefore use the following construction to determine the maximal noise threshold for simple successive decoding with a given decoding order μ.

Definition 3.2.3. Given a set of user specifications $\mathcal{U} = \{T_i, i \in [1, \ldots, N]\}$, and an ordering $\mu \in \mathbf{\Pi}(N)$, a set of disjoint power intervals $\{T'_i, i \in [1, \ldots, N]\}$ is said to be a p-*maximal allocation set with* μ if

- $\mathsf{p}(T'_i) = \mathsf{p}(T_i)$ for each $i \in [1, \dots, N]$;
- $T'_{\mu(N)} = T_{\mu(N)}$; and
- for each $i \in [2, \dots, N]$, the upper boundary of $T'_{\mu(i)}$ is the smaller of the upper boundary of $T_{\mu(i)}$ and the lower boundary of $T'_{\mu(i-1)}$, i.e.

$$\eta\left(T'_{\mu(i)}\right) + \mathsf{p}\left(T'_{\mu(i)}\right) = \min\left\{\eta\left(T_{\mu(i)}\right) + \mathsf{p}\left(T_{\mu(i)}\right), \eta\left(T'_{\mu(i-1)}\right)\right\} \tag{3.28}$$

Note that the last condition gives the upper boundary of $T'_{\mu(i)}$ as the minimum of the upper boundary of $T_{\mu(i)}$ and the lower boundary of $T'_{\mu(i-1)}$. Hence the p-maximal allocation set of a given set of users for any given μ may be constructed in a single pass from $T'_{\mu(1)}$ to $T'_{\mu(N)}$.

Also observe that (3.2.3) dictates that the p-maximal allocation set for any μ consists of disjoint power intervals, with $T'_{\mu(N)}$ being its lowest member.

Finally, note that (3.2.3) takes advantage of the observation discussed above, and $\eta\left(T'_{\mu(N)}\right)$ is the desired maximal noise threshold for simple successive decoding using μ. We have the following theorem.

Theorem 3.2.3. *The maximal noise threshold for simple successive decoding with a decoding order μ for a set of user specifications is given by the lower boundary of the lowest element of the set's* p*-maximal allocation set with μ.*

We have thus established that the maximal noise threshold for simple successive decoding using μ may be computed with $O(N)$ computations.

In (3.23), the maximal noise threshold for $\mathcal{U}$ to be SSD is formulated as maximium of the maximal noise threshold for simple successive decoding with a particular decoding order μ, over all possible decoding orders μ. For the convenience of the reader, we reproduce this formulation below.

$$\max_{\mu \in \mathbf{\Pi}(N)} \min_{i \in [1, \dots, N]} \eta\left(T_{\mu(i)}\right) - \sum_{j=i+1}^{N} \mathsf{p}\left(T_{\mu(j)}\right) \tag{3.29}$$

Since there is a total of $N!$ possible decoding orders μ, to achieve the objective of this subsection, which is to reduce the computational requirement to computing the maximal noise threshold for simple successive decoding to $O(N \ln N)$, we can not compute this quantity by first computing the maximal noise threshold for simple successive decoding using every every μ.

Note that one may alternatively consider this optimization as searching for the ordering μ that maximizes the maximal noise threshold for simple successive decoding using μ, which is the *optimal decoding order* that we mentioned at the beginning of this section. In the following, we will show that this optimal decoding order is the descending order of the upper boundaries of power intervals in $\mathcal{U}$. Specifically, we first offer an intuitive understanding of this fact. Then, we will prove this fact formally.

An intuitive understanding of this fact may be obtained by directly considering the question

> which user in a given set (or subset) is the optimal choice as the first decoded user?

Observe that once we know how to address this question, then we can solve the question of maximal noise threshold for simple successive decoding by incrementally finding the optimal choice of the next decoded user.

To address this question, observe that we have the following on the term in the optimization in (3.29)

$$\eta\left(T_{\mu(i)}\right) - \sum_{j=i+1}^{N} \mathsf{p}\left(T_{\mu(j)}\right) = \eta\left(T_{\mu(i)}\right) + \mathsf{p}\left(T_{\mu(i)}\right) - \sum_{j=i}^{N} \mathsf{p}(T_{\mu(j)}) \tag{3.30}$$

Note that the right side of (3.30) can be interpreted as

> moving the upper boundry of $T_{\mu(i)}$ to the left by an amount equal to the combined power specification of all the undecoded users, i.e. user $\mu(i)$ through $\mu(N)$

Since the term in (3.30) is first minimized in each μ and then maximized over all possible $\mu \in \mathbf{\Pi}(N)$ to obtain maximal noise threshold for simple successive decoding, one may alternatively approach the double optimization as finding the μ that maximizes the terms in the minimization. From this point of view, the above interpretation seems to suggest that the optimal user to be decoded next should be the one with the highest upper boundary (in the set of remaining undecoded users).

Definition 3.2.4. Given a set of user specifications $\mathcal{U}$, let μ^* be the ordering of power intervals in $\mathcal{U}$ according to the descending order of their upper boundaries[1]. Then the p-maximal allocation set for $\mathcal{U}$ with μ^* is called the p-*maximal allocation set* of $\mathcal{U}$, denote this set by $\mathsf{malloc}_{\mathsf{p}}(\mathcal{U})$.

We state and prove the main theorem of this subsection in the following.

Theorem 3.2.4. *The lower boundary of the lowest member of the* p-*maximal allocation set of a set of user specifications is the maximal noise threshold for the set to be SSD.*

[1] In case there exist multiple power intervals with identical upper boundaries, the ordering among these power intervals may be chosen arbitrarily.

Proof. We prove this theorem using induction.

First, we consider the 2-user case. Recall that the maximal noise threshold for the two users to be SSD is

$$\max_{\mu \in \mathbf{\Pi}(2)} \min \left\{\eta\left(T_{\mu(1)}\right) - \mathsf{p}(T_{\mu(2)}), \eta\left(T_{\mu(2)}\right)\right\} \tag{3.31}$$

Without the loss of generality, suppose the upper boundary of T_2 is less than that of T_1, i.e.

$$\eta(T_2) + \mathsf{p}(T_2) \leq \eta(T_1) + \mathsf{p}(T_1) \tag{3.32}$$

We show that the optimal decoding order is $\{1, 2\}$, and the desired maximal noise threshold quantity here is

$$\min \{\eta(T_1) - \mathsf{p}(T_2), \eta(T_2)\} \tag{3.33}$$

We divide our discussion into two cases depending on how the upper boundary of T_2 compares with the lower boundary of T_1. The proof for this case is divided into two cases.

In the first case, we suppose $\eta(T_2) + \mathsf{p}(T_2) \leq \eta(T_1)$, i.e. $T_2 < T_1$. Notice that the p-maximal allocation set with decoding order $\{2, 1\}$ for these two users is $\{T_1, T_2\}$. By Theorem 3.2.3, the maximal noise threshold for the two users to be SSD is $\eta(T_2)$ with this decoding order.

By Corollary 2.3.9, the maximal noise threshold for the two users to be achievable in this case is also $\eta(T_2)$. Since the achievability of the two users in the given channel is a necessary condition for the two to be SSD, we conclude that the maximal noise threshold for the two users to be SSD must not be smaller than $\eta(T_2)$. Therefore, the maximal noise threshold for the two users to be SSD with decoding order $\{1, 2\}$ must be the maximal noise threshold for the two users to be SSD. This completes the proof for this case.

Otherwise, we have

$$\eta(T_1) < \eta(T_2) + \mathsf{p}(T_2) \tag{3.34}$$

The p-maximal allocation set with decoding order $\{1, 2\}$ for these two users is $\{T_1, (\eta(T_1) - \mathsf{p}(T_2), \eta(T_1)]\}$. By Theorem 3.2.3, the maximal noise threshold for the two users to be SSD with this decoding order is $\eta(T_1) - \mathsf{p}(T_2)$.

Since $\mathsf{p}(T_2) > 0$, (3.32) implies that $\eta(T_2) - \mathsf{p}(T_1) < \eta(T_1)$. Hence, the p-maximal allocation set with the other decoding order, $\{2, 1\}$, for these two users is $\{(\eta(T_2) - \mathsf{p}(T_1), \eta(T_2)], T_2\}$. By Theorem 3.2.3, the maximal noise threshold for the two users to be SSD with this decoding order is $\eta(T_2) - \mathsf{p}(T_1)$.

Subtracting $\mathsf{p}(T_1) + \mathsf{p}(T_2)$ from both sides of (3.32), we have

$$\eta(T_2) - \mathsf{p}(T_1) \leq \eta(T_1) - \mathsf{p}(T_2) \tag{3.35}$$

This establishes the desired conclusion for this case, and hence completes the proof for the 2-user case.

To continue the induction, we assume the result of this theorem holds for all sets of user specifications with no greater than $N-1$ power intervals, we show the result for sets of N-user specifications below.

Let $\mathcal{U} = \{T_i = (a_i, b_i], i \in [1, \ldots, N]\}$. Without the loss of generality, we assume member power intervals of $\mathcal{U}$ are ordered according to their upper boundaries, i.e. $b_{i+1} \leq b_i$. Let $\mathsf{malloc_p}(\mathcal{U}) = \{T'_i = (a'_i, b'_i], i \in [1, \ldots, N]\}$. In particular, $\mathsf{p}(T'_i) = \mathsf{p}(T_i)$ Recall that $\mathcal{U}'$ consists of disjoint power intervals. We need to show that a'_N is the maximal noise threshold for $\mathcal{U}$ to be SSD. We show this in two cases.

First, suppose that $\mathsf{malloc_p}(\mathcal{U})$ consists of more than a single block of adjacent subsets. Let M be the largest i such that $\mathcal{V}' = \{T'_i, i \in [M, \ldots, N]\}$ is an adjacent set. By assumption, $M \leq N-1$ in this case. In particular, since b'_M is not a'_{M-1}, by definition of the p-maximal allocation set, $b'_M = b_M$. Let $\mathcal{V} = \{T_i, i \in [M, \ldots, N]\}$. Notice that $\mathcal{V}' = \mathsf{malloc_p}(\mathcal{V})$.

Since $\mathcal{V}$ consists of no greater than $N-1$ power intervals, by assumption, results of this theorem holds true for $\mathcal{V}$. In particular, a'_N is the maximal noise threshold for $\mathcal{V}$ to be SSD. Since adding more users into a multi-access channel can not ease the reception, the maximal noise threshold for $\mathcal{U}$ to be SSD can not be greater than that for $\mathcal{V}$. Since a'_N is also the maximal noise threshold for $\mathcal{U}$ to be SSD with decoding order $\{1, \ldots, N\}$, we have the desired conclusion for this case.

Otherwise, $\mathsf{malloc_p}(\mathcal{U})$ is an adjacent set. Let $\mathcal{V}' = \{T''_i, i \in 1, \ldots, N]\}$ be a p-maximal allocation set of $\mathcal{U}$ with some arbitrary ordering μ such that $\mathsf{p}(T''_i) = \mathsf{p}(T_i)$.

Since T_1 has the greatest of the upper boundaries in $\mathcal{U}$, the upper boundary of $T_{\mu(1)}$ is lower than that of T_1.

Note that, by the definition of p-maximal allocation set with μ, no member of $\mathcal{V}$ may have upper boundaries greater than that of $T_{\mu(1)}$. In addition, observe that, by the definition of p-maximal allocation set with μ, $\mathsf{p}(\mathsf{ext}(\mathsf{malloc_p}(\mathcal{U}))) = \mathsf{p}(\mathsf{ext}(\mathcal{V}'))$. Since, by assumption, $b_{\mu(1)} \leq b_N$, we conclude that

$$\eta(T''_{\mu(N)}) = b_{\mu(1)} - \mathsf{p}(\mathsf{ext}(\mathcal{V}')) \leq b_1 - \mathsf{p}(\mathsf{ext}(\mathsf{malloc_p}(\mathcal{U}))) = a'_N \tag{3.36}$$

Since μ is arbitrary, the proof is complete. □

Recall that we have observe that, for a given decoding order μ, members of the p-maximal allocation set of $\mathcal{U}$ with μ may be constructed one at a time following the decoding order in a single pass. We may therefore construct the p-maximal allocation set of $\mathcal{U}$ as in the following.

Algorithm 3.2.5.
Given a set of user specifications $\mathcal{U} = \{T_i = (a_i, b_i], i \in [1, \dots, N]\}$.

Step 1: *Order members of* $\mathcal{U}$ *according to their upper boundaries. Specifically, find* $\mu \in \mathbf{\Pi}(N)$ *such that* $b_{\mu(i+1)} \leq b_{\mu(i)}$ *for all feasible* i.

Step 2: *Construct the members of the* p*-maximal allocation set of* $\mathcal{U}$ *with the above* μ. *Specifically, initialize* $T'_{\mu(1)} = T_{\mu(1)}$. *Then for each* $i = 2$ *to* N, *compute*

$$b'_i = \min\left\{\eta\left(T_{\mu(i)}\right) + \mathsf{p}\left(T_{\mu(i)}\right), \eta\left(T'_{\mu(i-1)}\right)\right\}$$

and construct $T'_{\mu(i)} = \left(b_i - \mathsf{p}\left(T_{\mu(i)}\right), b_i\right]$.

The resulting $\{T'_i, i \in [1, \dots, N]\}$ *is the* p*-maximal allocation set of* $\mathcal{U}$, *and the maximal noise threshold for* $\mathcal{U}$ *to be SSD is* $\eta\left(T'_{\mu(N)}\right)$.

Observe that, since the construction of the p-maximal allocation set with μ in Step 2 takes exactly N steps, the total computational burden of this algorithm is upper bounded by the sort in Step 1. We therefore conclude that the maximal noise threshold for a set of N user specifications to be SSD is obtained requiring $O(N \ln N)$ computations. This accomplishes the main objective of our study on simple successive decoding.

By now, readers might notice the parallelism between the p-maximal allocation set construction used here to compute the maximal noise threshold for simple successive decoding and the irreducible equivalent construction used to compute the maximal noise threshold for achievability in Chapter 2. In the next subsection, we detail some of their similarities and differences. Recall, in Chapter 2, we presented an algorithm that computes the irreducible equivalent of a set incrementally, i.e. from the irreducible equivalent of a subset. As we will see, one result of such comparisons is an algorithm that computes the p-maximal allocation set incrementally.

3.2.3 The parallelism between the p-maximal allocation set and the irreducible equivalent

We now examine the constitution of the p-maximal allocation set, and present the near-perfect parallelism between this construction and the irreducible equivalent used in simplifying the

AWGN multi-access achievability conditions.

Recall that a p-maximal allocation set consists of disjoint power intervals. Observe that a set of disjoint power intervals may be partitioned into blocks of adjacent power intervals. For simplicity, we make the following definition.

Definition 3.2.5. For each $i \in [1, \ldots, M]$, let $\mathcal{V}_i$ be a set of adjacent power intervals with $\mathsf{ext}(\mathcal{V}_i)$ separated. Let $\mathcal{V} = \cup_{i=1}^{M} \mathcal{V}_i$. Then $\{\mathsf{ext}(\mathcal{V}_i), i \in [1, \ldots, M]\}$ is said to be an *outline set*, or *outline*, of $\mathcal{V}$. Denote it by $\mathsf{outline}(\mathcal{V}_i)$.

With this definition, observe that the outline of a p-maximal allocation set is a set of separated power intervals, each being the extent of a maximally adjacent subset of the p-maximal allocation set. Recall that the irreducible equivalent consists of separated power intervals, each being the combined user of an overlapping set. This is the first parallelism between the two constructions.

Secondly, the lower boundaries of the lowest power intervals in the outline of a p-maximal allocation set (which is equal to the lower boundary of the lowest power interval in the p-maximal allocation set) and the irreducible equivalent respectively designate the desired maximal noise threshold for simple successive decoding and for AWGN multi-access achievability.

We now examine the adjacent blocks in the p-maximal allocation set to reveal additional parallelism.

Definition 3.2.6. If a set of power intervals has an adjacent p-maximal allocation set, we say that this set of power intervals has the p-*overlapping* property.

Given $\mathcal{U} = \{T_1, T_2\}$. Observe that, by definition, the p-maximal allocation set of $\mathcal{U}$ is $\mathcal{U}$ when T_1 and T_2 are separated, and is an adjacent set otherwise. Hence, the p-overlapping property is identical to the overlapping property in the 2-user case. Moreover, the extent of their p-maximal allocation set when T_1 and T_2 overlap is a power interval with upper boundary at the greater of the upper boundaries of the two, and has the same power specification as their combined power constraint. Hence this extent must contain both power intervals. This is in parallel with the fact that the combined user of two overlapping power intervals contains both, which was established in Theorem 2.2.1.

Further resemblances between the two are revealed by the following two theorems.

Theorem 3.2.6. *Let $\mathcal{U}$ be a set of power intervals that has the* p-*overlapping property, and S be a power interval that does not intersect with* $\mathsf{ext}(\mathsf{malloc_p}(\mathcal{U}))$. *Then* $\mathsf{malloc_p}(\mathcal{U} \cup \{S\}) = \mathsf{malloc_p}(\mathcal{U}) \cup \{S\}$.

Proof. (This theorem follows directly from the definition of the p-maximal allocation set, and is omitted.) □

Theorem 3.2.7. *Let $\mathcal{U}$ be a set of power intervals that has the* p*-overlapping property, and S be a power interval that overlaps with* $\mathsf{ext}(\mathsf{malloc_p}(\mathcal{U}))$*. Then $\mathcal{U} \cup \{S\}$ has the* p*-overlapping property and the upper boundary of* $\mathsf{ext}(\mathsf{malloc_p}(\mathcal{U} \cup \{S\}))$ *equals the greater of the upper boundaries of S and* $\mathsf{ext}(\mathsf{malloc_p}(\mathcal{U}))$.

Proof. Let $\mathcal{U} = \{T_i = (a_i, b_i], i \in [1, \dots, N]\}$, and $\mathsf{malloc_p}(\mathcal{U}) = \{T'_i = (a'_i, b'_i], i \in [1, \dots, N]\}$ as in the above. Without the loss of generality, assume that members of $\mathcal{U}$ are ordered according to their upper boundaries such that $b_i < b_{i+1}$. We have $\mathsf{ext}(\mathsf{malloc_p}(\mathcal{U})) = (a'_1, b'_N]$. By definition of p-maximal allocation set, there is

$$b'_N = b_N \tag{3.37}$$

By assumption, $\mathsf{malloc_p}(\mathcal{U})$ is an adjacent set, i.e.

$$b'_i = a'_{i+1} \tag{3.38}$$

By definition of the p-maximal allocation set, $b'_i = \min\{a'_{i+1}, b_i\}$. Combining with (3.38), we have

$$a'_{i+1} \leq b_i \tag{3.39}$$

Let the additional power interval S be $(\alpha, \beta]$, and $\mathcal{U}' = \mathcal{U} \cup \{S\}$. Let $\mathsf{malloc_p}(\mathcal{U}') = \{T''_i = (a''_i, b''_i], i \in [1, \dots, N+1]\}$ such that $\mathsf{p}(T''_i) = \mathsf{p}(T_i)$ for $i \in [1, \dots, N]$, and $\mathsf{p}(T''_{N+1}) = \mathsf{p}(S)$. We show the corollary in three cases.

First, suppose the upper boundary of $\mathsf{ext}(\mathsf{malloc_p}(\mathcal{U}))$ is less than that of S, i.e. $b_N \leq \beta$. Therefore

$$b_i \leq \beta \tag{3.40}$$

By definition of the p-maximal allocation set, $T''_{N+1} = S$. Since S overlaps with $\mathsf{ext}(\mathsf{malloc_p}(\mathcal{U}))$, $\alpha \leq b_N$. Hence $b''_N = \min\{\alpha, b_N\} = \alpha$, and $a''_N = \alpha - \mathsf{p}(T_N)$. By (3.37), we have

$$b''_N \leq b'_N \tag{3.41}$$

In other words, the upper boundary of T''_N is below that of T'_N. Using the construction of the p-maximal allocation set in Algorithm 3.2.5, we conclude that $b''_i \leq b'_i$ for each $i \in [1, \dots, N]$, and hence $\mathsf{malloc_p}(\mathcal{U}')$ is disjoint. This completes the proof for this case.

Secondly, suppose the upper boundary of S is below that of $\mathsf{ext}(\mathsf{malloc_p}(\mathcal{U}))$, i.e. $\beta < b_1$.

By definition of p-maximal allocation set, $T''_i = T'_i$ for all $i \in [1, \ldots, N]$. Since S intersect with $\mathsf{ext}(\mathsf{malloc_p}(\mathcal{U}))$, $\beta > a'_1$. Therefore, $b''_{N+1} = \min\{a'_1, \beta\} = a'_1$. The desired conclusion follows for this case.

Otherwise, there exists an i with $i \in [1, \ldots, N-1]$ such that

$$b_i \leq \beta < b_{i+1} \tag{3.42}$$

By definition of p-maximal allocation set, $T''_j = T'_j$ for all $j \in [i+1, \ldots, N]$. Combining (3.39) and the first inequality in (3.42), we have $a'_{i+1} < \beta$. Hence

$$b''_{N+1} = \min\{a''_{i+1}, \beta\} = \min\{a'_{i+1}, \beta\} = a_{i+1} \tag{3.43}$$

Now, repeating the argument used in the first case, we arrive at the desired conclusion for this case, and the proof is complete. □

Observe that when $\mathcal{U}$ has the p-overlapping property, $\mathsf{p}(\mathsf{ext}(\mathsf{malloc_p}(\mathcal{U}))) = \mathsf{p}(\mathsf{T}(\mathcal{U}))$. Combining this fact when Theorem 3.2.7, we see that, when $\mathsf{malloc_p}(\mathcal{U})$ is adjacent, the extent of $\mathsf{malloc_p}(\mathcal{U} \cup \{S\})$ does not depend on the individual power intervals of $\mathcal{U}$. Therefore, whether a set of power intervals has the p-overlapping property can also be checked by successively replacing two overlapping sets with the extent of the p-maximal allocation set of the two, just like verifying the overlapping property. Hence we may apply the argument used in the 2-user case above repeatedly, and conclude that, when $\mathcal{U}$ has the overlapping property, the extent of $\mathsf{malloc_p}(()\mathcal{U})$ contains every power interval in $\mathcal{U}$.

We therefore have the following algorithm to construct $\mathsf{outline}(\mathsf{malloc_p}(\mathcal{U}))$.

Algorithm 3.2.8.
Let $\mathcal{U}' = \mathcal{U}$.

1. *If there exist T_1, $T_2 \in \mathcal{U}'$ such that T_1 and T_2 are overlapping (i.e. either intersecting or adjacent), then compute $\mathsf{malloc_p}(\{T_1, T_2\})$, and update*

$$\mathcal{U}' = \mathcal{U}' \setminus \{T_1, T_2\} \cup \{\mathsf{ext}(\mathsf{malloc_p}(\{T_1, T_2\}))\}$$

   and repeat this step

2. *Otherwise, stop*

The resulting $\mathcal{U}'$ is $\mathsf{outline}(\mathsf{malloc_p}(\mathcal{U}))$.

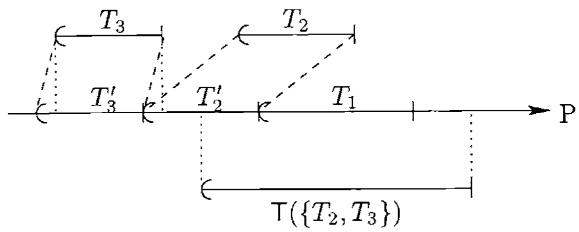


Figure 3-4: A set of 3 power intervals $\{T_i, i \in [1..3]\}$ which has the p-overlapping property, but not the overlapping property

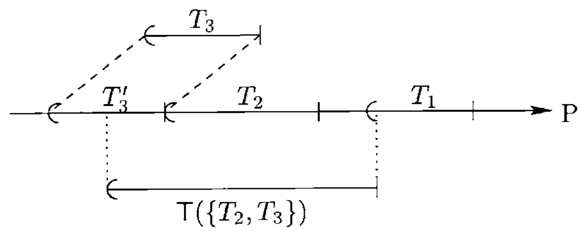


Figure 3-5: A set of 3 power intervals $\{T_i, i \in [1..3]\}$ which has the overlapping property, but not the p-overlapping property.

Notice that the only difference between Algorithm 3.2.8 and Algorithm 2.3.6 is the power interval that is used to replace the two overlapping power intervals in each iteration. Recall that the irreducible equivalent construction partitions a set of user specifications into maximally overlapping subsets, we similarly conclude that this construction partitions $\mathcal{U}$ into maximally p-overlapping blocks.

With this, we complete the presentation on the parallelism between the p-maximal allocation set and the irreducible equivalent construction.

Finally, we demonstrate the differences between the p-overlapping property and the overlapping property. Consider two overlapping power intervals $T_1 = (a_1, b_1]$ and $T_2 = (a_2, b_2]$ with $b_1 < b_2$. Let $b_1 \leq b_2$, we have $(a_2 - \mathsf{p}(T_1), b_2] = \mathsf{ext}(\mathsf{malloc_p}(\{T_1, T_2\}))$.

Let $T = (a, b] = \mathsf{T}(\{T_1, T_2\})$. By Theorem 2.2.1, there is

$$b > b_2 \tag{3.44}$$

$$\mathsf{r}((a, b]) = \mathsf{r}(T_1) + \mathsf{r}(T_2) \tag{3.45}$$

Hence $\mathsf{p}(\mathsf{ext}(\mathsf{malloc_p}(\{T_1, T_2\}))) = \mathsf{p}((a, b])$. Combining with (3.44), we conclude that $a' < a$. In other words, both the upper and the lower boundaries of the extent of the p-maximal allocation set are smaller than those of the combined user. As the result, a set of user specifications with the p-overlapping property may not have the overlapping property, neither does an overlapping set necessarily have the p-overlapping property. Figures 3-4 and 3-5 illustrate two counter examples with 3-user sets.

We end the discussion on simple successive decoding with a comment on the decoding order in the p-maximal allocation set. In the proof of Theorem 3.2.7, notice that the upper boundary of $\mathsf{ext}(\mathsf{malloc_p}(\mathcal{U} \cup \{S\}))$ is independent of the added user S when S has smaller upper boundary than $\mathsf{ext}(\mathcal{U})$. Hence, when $\mathsf{p}(S)$ is held fixed, moving S within the boundaries of $\mathsf{ext}(\mathcal{U})$ does not alter $\mathsf{ext}(\mathsf{malloc_p}(\mathcal{U} \cup \{S\}))$. Using the notation in the proof, suppose the upper boundary of S, β, satisfies $\beta \leq b'_j$ for some $j \in [1, \ldots, N]$. Let μ be a decoding order with S inserted just below T'_i, for all $0 < i \leq j$. Observe that maximal noise threshold for $\mathcal{U}$ to be SSD with any such μ is identical to that obtained with the p-maximal allocation set. In other words, multiple optimal decoding order may exist even when there is no ambiguity in the ordering of the power intervals in $\mathcal{U}$ according to their upper boundaries.

3.3 Power-Reduced Successive Decoding

As its name suggests, the power-reduction technique allows users to transmit at the same rate with less than the maximal power. It was first considered for multi-access channels where the transmission of some users may need to be decoded in the presence of interference from the transmission of other users. Therefore reducing the transmission power of selected users may improve the overall reception.

Definition 3.3.1. In an AWGN multi-access channel, a power interval T' is said to specify a *power-reduced user* of the power interval T if $\mathsf{r}(T') = \mathsf{r}(T)$ and $\mathsf{p}(T') < \mathsf{p}(T_2)$.

By the monotonicity of the $\mathsf{r}(\cdot)$ function, the lower boundary of T' must be no greater than that of T. Since the length (power specification) of T' is no greater than that of T, we conclude that the upper boundary of T' can not be greater than that of T'.

Theorem 3.3.1. *Let T' and T be two power intervals with $\mathsf{r}(T') = \mathsf{r}(T)$. Then T' is a power-reduced user of T if and only if both the upper and the lower boundaries of T' are no greater than those of T.*

Note that, by definition, a power-reduced user has the transitive property, i.e. if T_1 is a power-reduced user of T_2 and T_2 is a power-reduced user of T_3, then T_1 is a power-reduced user of T_3.

Definition 3.3.2. A set of user specifications is said to be a *power-reduced set* of a second set if there exists a one-to-one correspondence between the power intervals of the two sets, such that each user in the first set is the same or a power-reduced user of the corresponding user in the second set.

Similarly note that power-reduced sets also have the transitive property, i.e. if a set of users (or a set of power intervals) $\mathcal{U}_1$ is a power-reduced set of $\mathcal{U}_2$, and $\mathcal{U}_2$ is a power-reduced set of $\mathcal{U}_3$, then $\mathcal{U}_1$ is a power-reduced set of $\mathcal{U}_3$.

Recently, this technique was incorporated into code division multiple access (CDMA) cellular telephony practice. There, this technique is employed to uniformize the reception power of transmission signals for transmitters located at different distances from the receiver and has been called *power control*. As each user in this system is decoded regarding the transmission of all other users as interference, this practice improves the reception of weaker (in power) transmissions[2], and thus improves the overall system performance. To facilitate some theoretical comparison, observe that the AWGN multi-access achievable rate region that can be independently decoded is dominated by all the corner points on the dominant face. As illustrated in Figures 3-1 and 3-2, this region is inside the simple successive decoding region, and consists of rate tuples that are dominated by point O' in both figures.

To illustrate how the power-reduction technique may help enlarge the AWGN simple successive decoding region, consider two users specified by two intersecting power intervals $T_1 = (a_1, b_1]$ and $T_2 = (a_2, b_2]$ with $\mathsf{p}(T_1), \mathsf{p}(T_2) > 0$. Without the loss of generality, let $b_2 \leq b_1$. Observe that the p-maximal allocation set of these two power intervals is $\{T_1, (a_1 - \mathsf{p}(T_2), a_2]\}$, and their maximal noise threshold for simple successive decoding is $a_1 - \mathsf{p}(T_2)$. By Theorem 3.2.2, these two users are SSD in this channel if and only if the channel noise intensity n_o is below $a_1 - \mathsf{p}(T_2)$.

Suppose this is not the case, i.e.

$$a_1 - \mathsf{p}(T_2) < n_o \tag{3.46}$$

Then simple successive decoding is not available for this case. However, power-reduction may help to facilitate simple successive decoding to these two users in this case.

Recall, from the discussion of simple successive decoding above, that the two users may be SSD following the $\{2, 1\}$ order if and only if the gap between n_o and the lower boundary of T_1 is larger than the length (power constraint) of T_2. Consider reducing the transmission power (length) of user 2 to be smaller than the gap size, T_1 and the resulting power-reduced user of T_2 may be SSD in this channel. For example, suppose $T_2' = (n_o, b_2']$ is a power reduced user of T_2 for some $b_2' \in (n_o, a_1]$. Then $\{T_1, T_2'\}$ is SSD in this channel. This is illustrated in Figure 3-6.

To verify that such T_2' exists, note that, by definition of power-reduced user, $\mathsf{r}(T_2') = \mathsf{r}(T_2)$.

[2]Some of the causes leading to weaker transmission includes multipath-fading, and having the transmitter further away from the cell site.

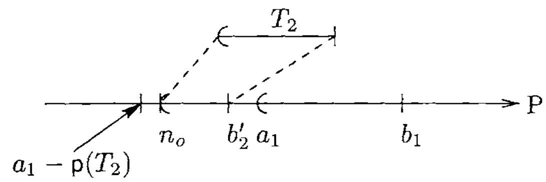


Figure 3-6: The gap between the lower boundary of T_1 and n_o must be big enough to contain a power interval with identical rate specification as T_2

Hence,

$$\frac{b_2'}{n_o} = \frac{b_2}{a_2} \tag{3.47}$$

Since $b_2' \leq a_1$ by assumption, we have $\frac{b_2}{a_2} \leq \frac{a_1}{n_o}$. In other words, power-reduction may be used to facilitate simple successive decoding for these two users with decoding order $\{1, 2\}$ if

$$\mathsf{r}(T_2) \leq \mathsf{r}((n_o, a_1]) \tag{3.48}$$

One important observation is that the power-reduced set has the same number of members as the original set. Hence, this technique requires the same number of decoding steps as simple successive decoding, which, as we will see, is about one-half the decoding steps required by either of the combination techniques studied in the next two chapters.

Definition 3.3.3. A set of user specifications $\mathcal{U}$ is said to be *decodable with simple successive decoding and power-reduction*, or *power-reduced successive decoding*, in an AWGN multi-access channel if there exists a power-reduced set of $\mathcal{U}$ that is SSD in the given channel. In such cases, $\mathcal{U}$ is said to be *PR-SD* in the channel.

In the above 2-user example, $\{T_1, T_2'\}$ is a power-reduced set of the two users that is SSD in the given channel. Note that in this example, reducing the power specification of user 2 further decreases its lower boundary to below the channel noise variance n_o, which renders its transmission undecodable. On the other hand, increasing the power specification of user 2 slightly above $\mathsf{p}(T_2') = \mathsf{p}((n_o, b_2'])$ does not violate the achievability conditions. In fact, $\{T_1, T_2'\}$ remains a power-reduced set of the two users that is SSD in the given channel so long as the power specification of T_2' does not exceed the gap size $a_1 - n_o$.

We call all the rate tuples in an AWGN multi-access achievable rate region that are power-reduced successive decoding collectively the *power-reduced successive decoding region* of the channel.

Incidentally, this discussion using two intersecting power intervals also shows that the power-reduced successive decoding region contains the simple successive decoding region for the two-user case. We will see that this is indeed true for N arbitrary users.

To avoid any misunderstanding, it is worthwhile to point out that in AWGN multi-access channels, there exist rate tuples outside of the simple successive decoding region that are successively decodable. This is because when the transmission rates of users decoded in later successive decoding steps are below their maxima, by the entropy power relations, having their transmission not appearing as white Gaussian processes (i.e. not averaging over the Gaussian ensemble of codes) can be used to increase the transmission rate of users decoded in earlier steps. Similarly, there also exist rate tuples outside of the power-reduced successive decoding region that are successively decodable with power-reduction. A proof of this is presented in Appendix D.

In light of this fact, we still chose to restrict our attention to successive decoding when users use the Gaussian ensemble of codes for the following three reasons.

1. The set of all successively decodable rate tuples (without any restriction on users' transmission codebooks) does not include all rate tuples in the AWGN multi-access achievable rate region. To see this, recall that the rate tuples on the dominant face achieve the maximal combined rate, and are therefore achievable only when all users use the Gaussian transmission codebooks. Since the simple successive decoding region does not include any rate tuples on the dominant face except the corner points, neither can the set of all successively decodable rate tuples.

2. Successive decoding with arbitrary user transmission codebooks may not help to reduce the overall decoding complexity since, as discussed in the introduction of this section, describing arbitrary transition probabilities for each successive decoding step may require large amounts of computation and/or storage elements. In comparison, restricting users to using the Gaussian ensemble of codes only requires the combined variance of the channel noise and the transmissions of the undecoded users to be computed for each decoding step, and therefore, at least in theory, leads to a dramatic simplification of the AWGN multi-access decoding complexity.

3. As will be seen in this and the next chapter, imposing this restriction admits the application of the power diagram framework into our analysis, which leads to understandings of, and constructions for, various successive decoding techniques requiring only near linear growth computations with the number of users in the system.

In the rest of this section, we establish an power diagram based $O(N \ln N)$ algorithm that tests whether a given set of N-user specifications is PR-SD in a given AWGN multi-access channel, and thereby provides a complete description of the power-reduced successive decoding region. Following the approach embedded in the power diagram framework, we address the question:

> What is the *maximal noise threshold* for a given set of user specifications to be PR-SD?

The rest of this section is organized as follows. We first formulate this problem in the power diagram framework in Subsection 3.3.1. Then, in Subsection 3.3.2, we develop an equivalent formulation, and show that it has the same structure as the one for simple successive decoding in (3.23). We therefore adapt the simplifying solution to the maximal noise threshold for simple successive decoding to simplify the solution to this problem. We then develop the duality between the simple successive decoding and the power-reduced successive decoding techniques in Subsection 3.3.3. There, we introduce the dual representation framework to the power diagram, which we call the *rate diagram*. Finally, we describe the set of all rate tuples that can be decoded with power-reduced successive decoding in Subsection 3.3.4.

3.3.1 The maximal noise threshold for power-reduced successive decoding

We formulate the question of maximal noise threshold for power-reduced successive decoding here in the power diagram framework.

We begin by formulating this problem for the two-user case in the power diagram framework. Let the two users be specified by two power intervals $T_1 = (a_1, b_1]$ and $T_2 = (a_2, b_2]$ with $\mathsf{p}(T_1), \mathsf{p}(T_2) > 0$. Without the loss of generality, we assume $b_2 \leq b_1$.

We do this in two cases. First, suppose T_1 and T_2 are separated. Since $b_2 < b_1$, we have $T_2 < T_1$. Observe that $\mathsf{malloc}_\mathsf{p}(\{T_1, T_2\}) = \{T_1, T_2\}$ in this case, hence the two users are SSD in any channel in which they are achievable. Hence the maximal noise threshold for these two users to be PR-SD is $\eta(T_2)$.

Otherwise, T_1 and T_2 overlap. From the discussion of power-reduced successive decoding with two intersecting power intervals in the introduction of this section, we have concluded that power-reduction may be used to facilitate simple successive decoding for these two users with decoding order $\{1, 2\}$ in a given channel with noise intensity n_o if (3.48) is satisfied. We now extend this result and show that these two users are PR-SD with this decoding order if and only if (3.48) is satisfied.

We do so by exhibiting the converse, i.e. if $\mathsf{r}((n_o, a_1]) < \mathsf{r}(T_2)$, then the two users may not be PR-SD with decoding order $\{1, 2\}$ in this channel. Suppose the contrary. Then, by definition, there exist T_1' and T_2', where T_1' is a power-reduced user of T_1 and T_2' is a power-reduced user of T_2, such that $\{T_1', T_2'\}$ is SSD in this channel with decoding order $\{2, 1\}$. By definition of maximal noise threshold for simple successive decoding, $\mathsf{p}(T_2') \leq \eta(T_1') - n_o$. From Theorem 3.3.1, $\eta(T_1') \leq \eta(T_1)$. Hence, we have

$$\mathsf{p}(T_2') \leq \eta(T_1) - n_o = a_1 - n_o \tag{3.49}$$

Since achievability is necessary for simple successive decoding, we conclude that $n_o < T_2'$. Combining with (3.49) and the monotonicity of the $\mathsf{r}(\cdot)$ function, we have $\mathsf{r}(T_2') \leq \mathsf{r}\left((n_o, a_1]\right)$. Since, by assumption, $\mathsf{r}\left((n_o, a_1]\right) < \mathsf{r}(T_2)$, we have $\mathsf{r}(T_2') < \mathsf{r}(T_2)$. This contradicts our assumption that T_2' is a power-reduced user of T_2. This completes the desired proof.

Now, condition (3.48) states that two overlapping power intervals T_1 and T_2 are PR-SD in the channel with decoding order $\{2,1\}$ if and only if the rate specification of the gap between the channel noise variance and the lower boundary of T_1 – the first decoded user – is greater than the rate of T_2. We will see in a moment that this can indeed be generalized to the general N-user case.

Combining results of the two cases above, we conclude that the two users are PR-SD in this channel with decoding order $\{1,2\}$ if and only if

$$n_o \leq \eta(T_2) \tag{3.50}$$

$$\mathsf{r}(T_2) \leq \mathsf{r}\left((n_o, \eta(T_1)]\right) \tag{3.51}$$

By definition of $\mathsf{r}(\cdot)$, (3.51) is equivalent to

$$n_o \leq \frac{\eta(T_1)}{\exp\left(2\,\mathsf{r}\,(T_2)\right)} \tag{3.52}$$

Hence, the maximal noise threshold for $\{T_1, T_2\}$ to be PR-SD with decoding order $\{1,2\}$ is

$$\max\left\{n_o :\; n_o \leq \eta(T_2),\; \mathsf{r}(T_2) \leq \mathsf{r}\left((n_o, \eta(T_1)]\right)\right\} = \min\left\{\eta(T_2), \frac{\eta(T_1)}{\exp\left(2\,\mathsf{r}\,(T_2)\right)}\right\} \tag{3.53}$$

Since the two users are PR-SD in a given channel if they are PR-SD with some decoding order, we conclude that the maximal noise threshold for $\{T_1, T_2\}$ to be PR-SD is

$$\begin{aligned} &\max_{\mu \in \mathbf{\Pi}(2)} \max\left\{n_o :\; n_o \leq \eta(T_{\mu(2)}),\; \mathsf{r}(T_{\mu(2)}) \leq \mathsf{r}\left((n_o, \eta(T_{\mu(1)})]\right)\right\} \\ &\quad = \max_{\mu \in \mathbf{\Pi}(2)} \min\left\{\eta(T_{\mu(2)}), \frac{\eta(T_{\mu(1)})}{\exp\left(2\,\mathsf{r}\left(T_{\mu(2)}\right)\right)}\right\} \end{aligned} \tag{3.54}$$

Before generalizing this formulation to N arbitrary users, we establish the following result.

Theorem 3.3.2. *Let $\mathcal{U}$ be a set of user specifications that is PR-SD using a given decoding* *order in an AWGN multi-access channel. Then there exists a disjoint power-reduced set of $\mathcal{U}$ with its members ordered according to the given decoding order which is achievable in the given channel.*

Remark. Note that by definition, the fact that $\mathcal{U}$ is PR-SD in the given channel implies that there exists a power-reduced set $\mathcal{V}$ of $\mathcal{U}$ that is SSD in the channel. However, as seen from the previous section, $\mathcal{V}$ may not consists of disjoint power intervals. This theorem establishes that there exists one such $\mathcal{V}$ consists of *disjoint* power intervals.

Proof. Let n_o denote the AWGN noise variance in the channel, and, without the loss of generality, let $\mu = \{1, \dots, N\}$ be the decoding order with which $\mathcal{U}$ is PR-SD in this channel.

By definition, there exists a power-reduced set $\mathcal{V}$ of $\mathcal{U}$ such that $\mathcal{V}$ is SSD with μ in this channel. To complete the proof, we construct a power-reduced set of $\mathcal{V}$ which consists of *disjoint* power intervals ordered according to μ which is achievable in this channel.

For $i \in [1, \dots, N]$, let T_i' be the power-reduced user of T_i in $\mathcal{V}$, i.e.

$$\mathsf{r}(T_i') = \mathsf{r}(T_i) \tag{3.55}$$

$$\mathsf{p}(T_i') \leq \mathsf{p}(T_i) \tag{3.56}$$

Since $\mathcal{V}$ is SSD with decoding order $[1, \dots, N]$ in the channel, by Theorem 3.2.1,

$$n_o \leq \min_{i \in [1, \dots, N]} \eta\left(T_i'\right) - \sum_{j=i+1}^{N} \mathsf{p}\left(T_j'\right) \tag{3.57}$$

In particular,

$$n_o + \sum_{j=i+1}^{N} \mathsf{p}\left(T_j'\right) \leq \eta\left(T_i'\right), \quad \forall i \in [1, \dots, N] \tag{3.58}$$

By the monotonicity of the $\mathsf{r}(\cdot)$ function, we have

$$\mathsf{r}(T_i') \leq \mathsf{r}\left(\left(n_o + \sum_{j=i+1}^{N} \mathsf{p}(T_j'), n_o + \sum_{j=i}^{N} \mathsf{p}(T_j')\right]\right), \quad \forall i \in [1, \dots, N] \tag{3.59}$$

In other words, for each $i \in [1, \dots, N]$, there exist

$$q_i \leq \mathsf{p}(T_i') \tag{3.60}$$

and

$$T_i'' = \left(n_o + \sum_{j=i+1}^{N} \mathsf{p}(T_j'), n_o + \sum_{j=i+1}^{N} \mathsf{p}(T_j') + q_i \right] \tag{3.61}$$

such that

$$\mathsf{r}\left(T_i''\right) = \mathsf{r}(T_i') \tag{3.62}$$

Observe that T_i'' is a power-reduced user of T_i'. Using the transitive property of power-reduced user, T_i'' is a power reduced user of T_i.

Finally, by (3.60), $T_i'' < T_{i-1}''$. Hence $\{T_i'', i \in [1, \dots, N]\}$ is a disjoint set. The proof is complete. □

Theorem 3.3.3. *A set of user specifications $\mathcal{U} = \{T_i, i \in [1, \dots, N]\}$ is PR-SD in a given channel with decoding order $\mu = \{\mu(1), \mu(2), \dots, \mu(N)\}$ if and only if for every $i \in [1, \dots, N]$, the combined rate specifications of users to be decoded following user $\mu(i)$ is no greater than the rate specification of the power interval between the channel noise intensity and the lower boundary of user $\mu(i)$'s power interval, i.e.*

$$\sum_{j=i+1}^{N} \mathsf{r}\left(T_{\mu(j)}\right) \leq \mathsf{r}\left(\left(n_o, \eta\left(T_{\mu(i)}\right)\right]\right) \tag{3.63}$$

Remark. Note that Condition 3.63 can be interpreted as stating that the combined rate specification of users to be decoded after the current user must be no greater than the rate specification of the gap between the channel noise intensity and the lower boundary of the current user's power interval, which is a simple generalization of Condition 3.51 for the 2-user case.

Proof. Let n_o denote the variance of the channel noise. Without the loss of generality, let $\mu = \{1, \dots, N\}$. Specifically, for each $i \in [1, \dots, N]$, let user i's transmission be decoded only with the decoding results of users 1 through $i-1$, and in the presence of the channel noise and the combined interference from transmissions of power-reduced users $i+1$ through N.

For the forward direction, we assume (3.63), and need to show that $\mathcal{U}$ is PR-SD in this channel.

First consider (3.63) for $i = N$, i.e. $0 \leq \mathsf{r}\left(\left(n_o, \eta\left(T_N\right)\right]\right)$. By definition of $\mathsf{r}(\cdot)$, this is equivalent to $n_o < T_N$. Hence, there exist $q_N \leq \mathsf{p}(T_N)$ and $T_N' = (n_o, n_o + q_N]$ such that $\mathsf{r}(T_N') = \mathsf{r}(T_N)$. By definition, T_N' is a power-reduced user of T_N.

Now, consider (3.63) for $i = N-1$, i.e. $\mathsf{r}(T_N) \leq \mathsf{r}\left(\left(n_o, \eta\left(T_{N-1}\right)\right]\right)$. Combining with the fact that T'_N is a power-reduced user of T_N, we have

$$\mathsf{r}(T'_N) = \mathsf{r}\left((n_o, n_o + q_N]\right) \leq \mathsf{r}\left((n_o, \eta\left(T_{N-1}\right)]\right) \tag{3.64}$$

The inequality in the above is equivalent to $n_o + q_N < T_{N-1}$. Hence, there exist $q_{N-1} \leq \mathsf{p}(T_{N-1})$ and $T'_{N-1} = (n_o + q_N, n_o + q_N + q_{N-1}]$ such that $\mathsf{r}(T'_{N-1}) = \mathsf{r}(T_{N-1})$. By definition, T'_{N-1} is a power-reduced user of T_{N-1}.

Repeating this construction for each $i = N-2$ down to $i = 1$, we see that $\mathcal{V} = \{T'_i, i \in [1, \ldots, N]\}$ is a power-reduced set of $\mathcal{U}$, and $\mathcal{V}$ consists of adjacent power intervals with $T'_i \leq T'_{i-1}$ for all suitable i. By Corollary 2.3.9, the maximal noise threshold for $\mathcal{V}$ to be achievable is n_o. Thus $\mathcal{V}$ is achievable in the given channel.

By the definition of the p-maximal allocation set, $\mathsf{malloc}_{\mathsf{p}}(\mathcal{V}) = \mathcal{V}$. Therefore, $\mathcal{V}$ is decodable with simple successive decoding in every channel in which it is achievable. Since we have concluded that $\mathcal{V}$ is achievable in the given channel, the proof for this part is complete.

The converse is proven following essentially the same logic as in the two-user case above. Specifically, let the greatest index in $[1, \ldots, N]$ for which (3.63) is violated be i^*, i.e.

$$\sum_{j=i^*+1}^{N} \mathsf{r}(T_j) > \mathsf{r}\left((n_o, \eta\left(T_{i^*}\right)]\right) \tag{3.65}$$

Suppose that this set of users is PR-SD in the given channel. By Theorem 3.3.2, there exists a power-reduced set $\mathcal{V}$ that is disjointly ordered according to the decoding order, and is achievable in the given channel. Let T'_i be the power-reduced user of T_i in $\mathcal{V}$. We have $T'_i < T'_{i-1}$ and $\mathsf{r}(T'_i) = \mathsf{r}(T_i)$. In particular,

$$\mathsf{ext}\left(\left\{T'_j, j' \in [i^* + 1, \ldots, N]\right\}\right) \subseteq \left(n_o, \eta(T'_{i^*})\right] \tag{3.66}$$

Since T'_{i^*} is a power-reduced user of T_{i^*}, by Theorem 3.3.1, $\eta(T'_{i^*}) \leq \eta(T_{i^*})$. Combining with (3.66), we have

$$\mathsf{ext}\left(\left\{T'_j, j' \in [i^* + 1, \ldots, N]\right\}\right) \subseteq \left(n_o, \eta(T_{i^*})\right] \tag{3.67}$$

Finally, since $\mathcal{V}$ is disjoint,

$$\sum_{j=i^*+1}^{N} \mathsf{r}(T'_j) \leq \mathsf{r}\left(\mathsf{ext}\left(\left\{T'_j, j' \in [i^* + 1, \ldots, N]\right\}\right)\right) \leq \mathsf{r}\left((n_o, \eta(T_{i^*})]\right) \tag{3.68}$$

Combining with (3.65), we have

$$\sum_{j=i^*+1}^{N} \mathsf{r}(T_j') < \sum_{j=i^*+1}^{N} \mathsf{r}(T_j) \tag{3.69}$$

This contradicts the assumption that T_i' is a power-reduced user of T_i for each $i \in [1, \ldots, N]$. This completes the proof of the converse. □

Note that each condition in (3.63) may be interpreted as requiring that, for each $i \in [1, \ldots, N]$, the rate specifications between the channel noise variance and the lower boundary of user i's power interval must be greater than the combined rate of all the users to be decoded after user i.

By definition of the $\mathsf{r}(\cdot)$ function, (3.63) is equivalent to

$$n_o \leq \frac{\eta\left(T_{\mu(i)}\right)}{\exp\left(2\sum_{j=i+1}^{N} \mathsf{r}\left(T_{\mu(j)}\right)\right)} \tag{3.70}$$

Definition 3.3.4. Given a set of user specifications $\mathcal{U} = \{T_i, i \in [1, \ldots, N]\}$ and a decoding order $\mu \in \mathbf{\Pi}(N)$, the *maximal noise threshold for $\mathcal{U}$ to be PR-SD with μ* is defined to be

$$\begin{aligned} &\max\left\{n_o : \sum_{j=i+1}^{N} \mathsf{r}\left(T_{\mu(j)}\right) \leq \mathsf{r}\left(\left(n_o, \eta\left(T_{\mu(i)}\right)\right]\right),\ \forall i \in [1, \ldots, N]\right\} \\ &= \min_{i\in[1,\ldots,N]} \frac{\eta\left(T_{\mu(i)}\right)}{\exp\left(2\sum_{j=i+1}^{N} \mathsf{r}\left(T_{\mu(j)}\right)\right)} \end{aligned} \tag{3.71}$$

Since a set of user specifications is PR-SD in a channel if it is PR-SD with at least one decoding order, we have the following.

Definition 3.3.5. The *maximal noise threshold for a set of user specifications* $\{T_i, i \in [1, \ldots, N]\}$ *to be PR-SD* is defined to be

$$\begin{aligned} &\max_{\mu\in\mathbf{\Pi}(N)}\left\{\max\left\{n_o : \sum_{j=i+1}^{N} \mathsf{r}\left(T_{\mu(j)}\right) \leq \mathsf{r}\left(\left(n_o, \eta\left(T_{\mu(i)}\right)\right]\right),\ \forall i \in [1, \ldots, N]\right\}\right\} \\ &= \max_{\mu\in\mathbf{\Pi}(N)} \min_{i\in[1,\ldots,N]} \frac{\eta\left(T_{\mu(i)}\right)}{\exp\left(2\sum_{j=i+1}^{N} \mathsf{r}\left(T_{\mu(j)}\right)\right)} \end{aligned} \tag{3.72}$$

The discussion above provides the following theorem.

Theorem 3.3.4. *A set of user specifications is PR-SD in an AWGN multi-access channel if* *and only if the channel noise variance is below the maximal noise threshold for the set to be PR-SD.*

Since there are $N!$ distinct decoding orders for a set of N users, a brute force computation of this maximal noise threshold quantity using (3.72) requires a multitude of $N!$ computations. In the next subsection, we develop an algorithm to compute it through a construction requiring $O(N \ln N)$ computations.

3.3.2 Computing the maximal noise threshold for power-reduced successive decoding with the r-maximal allocation set

In this section, we re-formulate the maximal noise threshold for power-reduced successive decoding to have identical structure as the formulation of the maximal noise threshold for simple successive decoding. Consequently, the simplified solution developed for the latter in Subsection 3.2.2 is a simplified solution to the former.

Since the logarithm is a monotonically increasing function, we may equivalently consider the $\frac{1}{2}\ln(\cdot)$ of the terms inside the right-hand side of (3.71). Specifically, we consider the following equivalent minimization.

$$\min_{i\in[1,\cdots,N]} \frac{1}{2}\ln\left(\eta\left(T_{\mu(i)}\right)\right) - \sum_{j=1}^{i-1} \mathsf{r}\left(T_{\mu(j)}\right) \tag{3.73}$$

The outcome of this minimization is one-half the logarithm of the desired maximal noise threshold for $\mathcal{U}$ to be PR-SD for μ. We therefore have the following equivalent formulation of maximal noise threshold for power-reduced successive decoding.

$$\max_{\mu\in\mathbf{\Pi}(N)} \min_{i\in[1,\cdots,N]} \frac{1}{2}\ln\left(\eta\left(T_{\mu(i)}\right)\right) - \sum_{j=1}^{i-1} \mathsf{r}\left(T_{\mu(j)}\right) \tag{3.74}$$

Recall that, in (3.22), the maximal noise threshold for $\mathcal{U}$ to be SSD for μ is formulated as

$$\min_{i\in[1,\cdots,N]} \eta\left(T_{\mu(i)}\right) - \sum_{j=1}^{i-1} \mathsf{p}\left(T_{\mu(j)}\right) \tag{3.75}$$

Note that if we replace the lower boundary of $T_{\mu(i)}$ by one-half its logarithm, and replace the power specifications of power intervals to be decoded after user $\mu(i)$ by their rate specifications, we arrive at the formulation of maximal noise threshold for $\mathcal{U}$ to be PR-SD in (3.73). Hence, we obtain the following simplifying solution to maximal noise threshold for $\mathcal{U}$ to be PR-SD by similarly replacing these terms in the solution method developed for simplifying the computation of the maximal noise threshold for $\mathcal{U}$ to be SSD for μ.

Definition 3.3.6. Given a set of user specifications $\mathcal{U} = \{T_i, i \in [1, \dots, N]\}$, and an ordering $\mu \in \mathbf{\Pi}(N)$, a set of disjoint power intervals $\{T'_i, i \in [1, \dots, N]\}$ is said to be a r-*maximal allocation set with* μ if

- $\mathsf{r}(T'_i) = \mathsf{r}(T_i)$ for each $i \in [1, \dots, N]$;
- $T'_{\mu(1)} = T_{\mu(1)}$; and
- for each $i \in [2, \dots, N]$, the upper boundary of $T'_{\mu(i)}$ is the smaller of the upper boundary of $T_{\mu(i)}$ and the lower boundary of $T'_{\mu(i-1)}$, i.e.

$$\begin{aligned} &\frac{1}{2}\ln\left(\eta\left(T'_{\mu(i)}\right) + \mathsf{p}\left(T'_{\mu(i)}\right)\right) \\ &\quad = \min\left\{\frac{1}{2}\ln\left(\eta\left(T_{\mu(i)}\right) + \mathsf{p}\left(T_{\mu(i)}\right)\right), \frac{1}{2}\ln\left(\eta\left(T'_{\mu(i-1)}\right) + \mathsf{p}\left(T'_{\mu(i-1)}\right)\right) - \mathsf{r}\left(T'_{\mu(i-1)}\right)\right\} \end{aligned} \tag{3.76}$$

By definition, $\mathsf{r}(T)$ is the difference between one-half the logarithm of its upper and lower boundaries. Letting its upper boundary be b, we have

$$\mathsf{r}(T) = \frac{1}{2}\ln(b) - \frac{1}{2}\ln(\eta(T)) \tag{3.77}$$

Hence

$$\frac{1}{2}\ln(\eta(T)) = \frac{1}{2}\ln(b) - \mathsf{r}(T) \tag{3.78}$$

In particular,

$$\frac{1}{2}\ln\left(\eta\left(T'_{\mu(i-1)}\right) + \mathsf{p}\left(T'_{\mu(i-1)}\right)\right) - \mathsf{r}\left(T'_{\mu(i-1)}\right) = \frac{1}{2}\ln\left(\eta\left(T'_{\mu(i-1)}\right)\right) \tag{3.79}$$

Therefore, we have the following equivalent to (3.76):

$$\eta\left(T'_{\mu(i)}\right) + \mathsf{p}\left(T'_{\mu(i)}\right) = \min\left\{\eta\left(T_{\mu(i)}\right) + \mathsf{p}\left(T_{\mu(i)}\right), \eta\left(T'_{\mu(i-1)}\right)\right\} \tag{3.80}$$

Combining with the requirement that $\mathsf{r}(T'_i) = \mathsf{r}(T_i)$ for each $i \in [1, \dots, N]$, we conclude that the r-maximal allocation set of $\mathcal{U}$ for μ is a power-reduced set of $\mathcal{U}$.

Following the same reasoning as in the proof of Theorem 3.2.3, we have the following theorem.

Theorem 3.3.5. *The maximal noise threshold for power-reduced successive decoding with a decoding order* μ *for a set of user specifications is given by the lower boundary of the lowest element of the set's* r*-maximal allocation set with* μ.

Since logarithm preserves ordering, we follow the development for simplifying the computation of maximal noise threshold for $\mathcal{U}$ to be SSD and define the following.

Definition 3.3.7. Given a set of user specifications $\mathcal{U}$, let μ^* be the ordering of power intervals in $\mathcal{U}$ according to the ascending order of their upper boundaries[3]. Then the r-maximal allocation set for $\mathcal{U}$ with μ^* is called the r*-maximal allocation set* of $\mathcal{U}$, denoted by $\mathsf{malloc}_{\mathsf{r}}(\mathcal{U})$.

Following the same reasoning as in the proof of Theorem 3.2.4, we have the following.

Theorem 3.3.6. *The lower boundary of the lowest member of the* r*-maximal allocation set of a set of user specifications is the maximal noise threshold for the set to be PR-SD.*

In particular, the r-maximal allocation set of a set of user specifications $\mathcal{U}$ is a power-reduced set of $\mathcal{U}$ that is SSD in any channel in which $\mathcal{U}$ is PR-SD.

Similarly, we have the following algorithm for constructing the r-maximal allocation set.

Algorithm 3.3.7.
Given a set of user specifications $\mathcal{U} = \{T_i = (a_i, b_i], i \in [1, \dots, N]\}$.

Step 1: *Order members of* $\mathcal{U}$ *according to their upper boundaries. Specifically, find* $\mu \in \mathbf{\Pi}(N)$ *such that* $b_{\mu(i-1)} \leq b_{\mu(i)}$ *for all feasible* i

Step 2: *Construct the members of* r*-maximal allocation set of* $\mathcal{U}$ *with* μ *following* μ. *Specifically, initialize* $T'_{\mu(N)} = T_{\mu(N)}$. *Then for each* $i = N-1$ *down to* 1, *compute* $b = \min\{\eta(T_{\mu(i)}) + \mathsf{p}(T_{\mu(i)}), \eta(T_{\mu(i+1)'})\}$, *and construct* $T'_{\mu(N)} = (a, b]$ *such that*

$$\mathsf{r}\left(T'_{\mu(N)}\right) = \frac{1}{2}\ln\left(\frac{b}{a}\right) = \mathsf{r}\left(T_{\mu(N)}\right) \tag{3.81}$$

The resulting $\{T'_i, i \in [1, \dots, N]\}$ *is the* r*-maximal allocation set of* $\mathcal{U}$, *and the maximal noise threshold for* $\mathcal{U}$ *to be PR-SD is* $\eta(T'_{\mu(1)})$.

We similarly conclude that the maximal noise threshold for a set of N user specifications to be PR-SD is obtained using at most $O(N \ln N)$ computations. This accomplishes the main objective of our study on power-reduced successive decoding.

[3] Again, in case there exist multiple power intervals with identical upper boundaries, the ordering among these power intervals may be chosen arbitrarily.

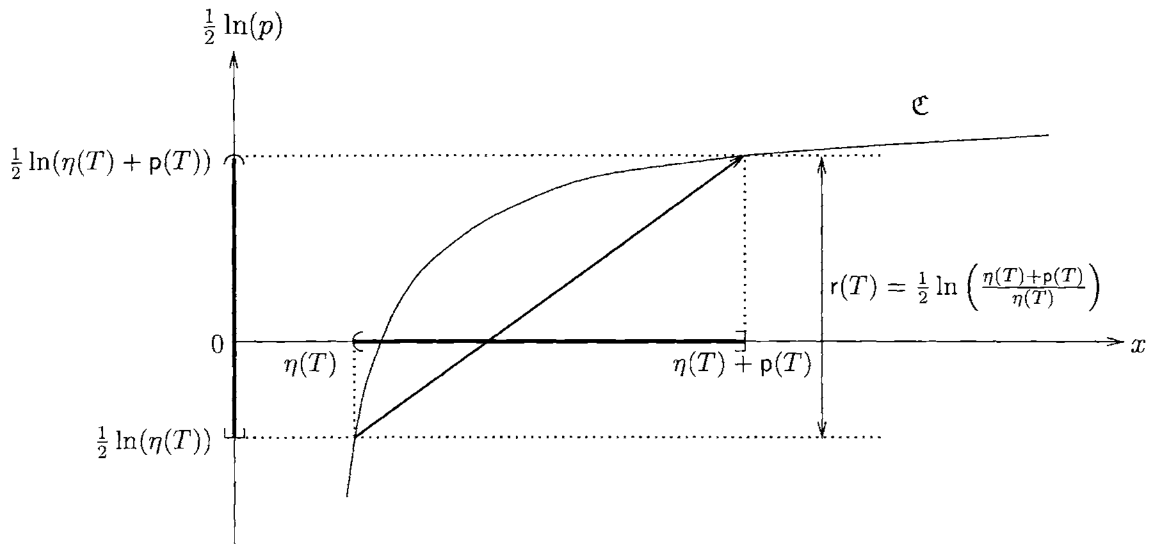


Figure 3-7: The duality between the power interval and the rate interval representations

3.3.3 The duality between simple successive decoding and power-reduced successive decoding

In this subsection, we show that the similarities between simple successive decoding and power-reduced successive decoding goes beyond just the formulations and solutions for their respective maximal noise threshold problems. In the following, we first present a dual representation framework to the power diagram, which we call the *rate diagram.* We then show that the two successive decoding methods discussed above are dual to each other in these two frameworks.

We start with the definition of the rate interval representation of user specifications.

Definition 3.3.8. Let a user be specified by its power constraint p and maximal noise threshold η. Then, the user's *rate interval* is defined to be the left-open-right-closed interval $(\frac{1}{2}\ln(\eta), \frac{1}{2}\ln(\eta+p)]$ on the real axis.

The real axis on which rate intervals are placed is called the *rate axis.*

Observe that the lower and upper boundaries of a user's rate interval are respectively one-half the logarithm of the lower and upper boundaries of a user's power interval.

Note that the power interval and the rate interval of a user are dual representations of each other when viewed through the universal function $\mathfrak{C}$. Specifically, the power interval and the rate interval are interval representations of the user specifications on the power and rate axis of the universal function $\mathfrak{C}$ respectively. This is depicted in Figure 3-7.

Let the outcome of a mathematical function, such as logarithm and exponentiation, with a set of intervals on the real axis as inputs be the union of the outcome of the function on each member of the set. We may therefore denote the rate interval of a power interval T as $\frac{1}{2}\ln(T)$.

Similarly, the power interval of a rate interval T_r is $\exp(2T_r)$. This establishes a translation (through the universal curve $\mathfrak{C}$) for the interval representations of users from the power diagram to the rate diagram.

For convenience, we make the following definitions.

Definition 3.3.9. Given a user's rate interval $T_r = (a, b]$,

- $\eta_r(T_r) \equiv a$ is the lower boundary of the rate interval, which, by definition of rate interval, is one-half the logarithm of the user's maximal noise threshold.
- $\mathsf{p}_r(T_r) \equiv \mathsf{p}(\exp(2T_r)) = \exp(2b) - \exp(2a)$ is the user's power constraint.
- $\mathsf{r}_r(T_r) \equiv \mathsf{r}(\exp(2T_r)) = \frac{1}{2}(\ln(\eta(\exp(2T)) + \mathsf{p}(\exp(2T))) - \ln(\eta(\exp(2T)))) = b - a$ is the user's transmission rate specification.

Through the translation mechanism introduced above, we can therefore place multiple rate intervals on the rate axis, just like placing multiple power intervals on the power axis. We call this construct the *rate diagram*.

Furthermore, this translation mechanism can be used to directly translate all results established in this document from one framework to the other. In the following, we detail the three aspects of this translation concerning the manipulations of power intervals.

First, since the $\mathfrak{C}$ function preserves ordering, i.e. x_1 is respectively less than, equal to, or greater than x_2 if and only if $\frac{1}{2}\ln(x_1)$ is less than, equal, or greater than $\frac{1}{2}\ln(x_2)$, two power intervals T_1 and T_2 are respectively intersecting, adjacent, or separated if and only if their rate intervals are intersecting, adjacent, or separated.

Secondly, consider the following definition of the extent in the rate diagram.

Definition 3.3.10. The *extent* of a set of rate intervals $\mathcal{U}_r$ is the smallest rate interval (in length) that contains all member rate intervals in the set, and is denoted $\mathsf{ext}_r(\mathcal{U}_r)$.

Hence, $\mathsf{ext}_r(\mathcal{U}_r) = \mathsf{ext}(\{\exp(2T_r), T_r \in \mathcal{U}_r\})$. In other words, the definition of the extent is consistent over the two representation frameworks.

Finally, by definition, the combined user of a set of users $\mathcal{V}$ is specified by the combined power and rate specifications of users in $\mathcal{V}$, we see that the rate interval of the combined specification of $\mathcal{V}$ is $\frac{1}{2}\ln(\mathsf{T}(\mathcal{V}))$.

Combining these three aspects, we see that as far as representations and manipulations of user specifications in AWGN multi-access channels are concerned, the power diagram framework and the rate diagram framework are in fact equivalent.

We have chosen to work primarily with the power diagram framework in this document because of two reasons. One, the non-negativity of the boundaries of power intervals is more intuitive, whereas the boundaries of rate intervals can be negative; and two, the power diagram framework better suits most of the problems (except this one on power-reduced successive decoding).

We now discuss the duality between simple successive decoding and power-reduced successive decoding techniques. This duality presents itself in three aspects.

First, recall that each user is required to transmit at full power in simple successive decoding, whereas each user is required to transmit at the exact rate specification in power-reduced successive decoding.

Secondly, consider interpreting the formulation of the maximal noise threshold for power-reduced successive decoding above using the rate intervals. In particular, we may represent each condition in (3.72) in the rate diagram as

$$\sum_{j=1}^{i-1} \mathsf{r}\left(T_{\mu(j)}\right) \leq \mathsf{r}_{\mathsf{r}}\left(\frac{1}{2}\ln\left(\left(n_o, \eta\left(T_{\mu(i)}\right)\right]\right)\right) = \eta_{\mathsf{r}}\left(\frac{1}{2}\ln\left(T_{\mu(i)}\right)\right) - \frac{1}{2}\ln\left(n_o\right) \tag{3.82}$$

Since $\frac{1}{2}\ln(n_o)$ is the channel noise intensity in the rate diagram, this condition may thus be interpreted as requiring the gap, on the rate axis, between the noise intensity and the lower boundary of the rate interval of $T_{\mu(i)}$ to be greater than the combined rate specification of users to be decoded after user $\mu(i)$. Recall that conditions in the formulation of the maximal noise threshold for simple successive decoding in (3.23) have identical interpretation on the power axis.

Finally, recall that we have adapted the simplifying solution to the maximal noise threshold for simple successive decoding to simplify the computation of the maximal noise threshold for power-reduced successive decoding because the two have identical structure in their formulations. Indeed, this adaptation simply converted the construction for the p-maximal allocation set in the power diagram to the rate diagram. This can be seen from interpreting the steps in Algorithm 3.3.7) in the rate diagram. In particular, since a user's rate interval is one-half the logarithm of its power interval, the ordering using the upper boundaries of rate intervals is identical to that of the corresponding power intervals. Hence the resulting ordering in Step 1 is identical in either representation framework. In Step 2, the upper boundary of $T'_{\mu(i)}$ (which is b) is taken to be the smaller of the upper boundary of $T_{\mu(i)}$ and the lower boundary of $T'_{\mu(i+1)}$. This may be equivalently stated as letting the upper boundary of the rate interval of $T'_{\mu(i)}$ to be the smaller of the upper boundary of the rate interval of $T_{\mu(i)}$ and the lower boundary of the rate interval of $T_{\mu(i+1)}$. The condition (2) can similarly be stated using the rate intervals as

$$\mathsf{r}_\mathsf{r}\left(\frac{1}{2}\ln\left(T'_{\mu(N)}\right)\right) = \mathsf{r}_\mathsf{r}\left(\frac{1}{2}\ln\left(T_{\mu(N)}\right)\right) \tag{3.83}$$

By definition of rate interval, we see that the lower boundary of the rate interval of $T'_{\mu(i}$ must be

$$\frac{1}{2}\ln(b) - \mathsf{r}_\mathsf{r}\left(\frac{1}{2}\ln\left(T_{\mu(N)}\right)\right) = \frac{1}{2}\ln(b) - \mathsf{r}\left(T_{\mu(N)}\right) \tag{3.84}$$

Observe that this construction is identical to the construction on the power axis used in Algorithm 3.2.5.

This completes the discussion on the duality of these two successive decoding techniques.

Before leaving this subsection, let us note that identical parallelism exists between the r-maximal allocation set and the irreducible equivalent construction as the one presented between the p-maximal allocation set and the irreducible equivalent construction in Subsection 3.2.3.

3.3.4 The PR-SD rate region

In an N-user AWGN multi-access channel with channel noise variance n_o, let $\mathsf{P} = \{p_i, i \in [1..N]\}$ be the power specifications of the users. In this channel, let the power-reduced successive decoding rate region be denoted by $\mathfrak{D}(\mathsf{P})$. In this section, we develop a characterization of $\mathfrak{D}(\mathsf{P})$ using results in the previous subsections.

Consider a rate tuple $\mathsf{R} = \{r_i, i \in [1..N]\}$ in $\mathfrak{D}(\mathsf{P})$. Note that the set of user specifications $\{(p_i, r_i), i \in [1..N]\}$ is power-reduced successive decoding in the given channel. Let $\mathcal{U}(\mathsf{R}) = \{T_i(r_i), i \in [1..N]\}$ denote this set of user specifications using the power interval representation. In particular

$$\mathsf{r}\left(T_i(r_i)\right) = r_i \tag{3.85}$$

$$\mathsf{p}\left(T_i(r_i)\right) = p_i \tag{3.86}$$

Using Theorem 3.3.2, we see that $\mathfrak{D}(\mathsf{P})$ is the collection of all rate tuples, each with a r-maximal allocation set of $\mathcal{U}(\mathsf{R})$ that is achievable in the given channel.

Recall that the r-maximal allocation set set of $\mathcal{U}(\mathsf{R})$ consists of disjoint power intervals, each being a power-reduced user of one user in $\mathcal{U}(\mathsf{R})$. Hence, one may classify $\mathfrak{D}(\mathsf{P})$ according to the decoding order of the r-maximal allocation set of the resulting $\mathcal{U}(\mathsf{R})$. Specifically, for each

$\mu \in \mathbf{\Pi}(N)$, let $\mathfrak{A}(\mathsf{P},\mu)$ denote the following collection of sets of disjoint power intervals

$$\mathfrak{A}(\mathsf{P},\mu) = \left\{\{S_i, i \in [1..N]\} : n_o < S_{\mu(1)} < S_{\mu(2)} < \ldots < S_{\mu(N)}, \text{ and } \mathsf{p}(S_i) \leq \mathsf{p}(T_i), \forall i \in [1..N]\right\} \tag{3.87}$$

and let $\mathfrak{R}(\mathsf{P},\mu)$ denote the set of all rate tuples that are achieved by members of $\mathfrak{A}(\mathsf{P},\mu)$, i.e.

$$\mathfrak{R}(\mathsf{P},\mu) = \{(\mathsf{r}(S_1), \mathsf{r}(S_2), \ldots, \mathsf{r}(S_N)), \forall \{S_i, i \in [1..N]\} \in \mathfrak{A}(\mathsf{P},\mu)\} \tag{3.88}$$

Then

$$\mathfrak{D}(\mathsf{P}) = \bigcup_{\forall \mu \in \mathbf{\Pi}(N)} \mathfrak{R}(\mathsf{P},\mu) \tag{3.89}$$

Observe that this union is in general not convex, as it contains every corner point on the dominant face of the AWGN multi-access achievable rate region, but not the rest of the the dominant face.

Also recall that every corner point on the dominant face of the AWGN multi-access achievable rate region can only be successively decoded with a single decoding order. Hence each such corner point belongs to a unique $\mathfrak{R}(\mathsf{P},\mu)$. Therefore, the union in (3.89) must be taken over all μ.

Let $\mathsf{Q} = \{q_i, i \in [1, \ldots, N]\}$ be a set of power specifications with

$$q_i \leq p_i, \quad \forall i \in [1, \ldots, N] \tag{3.90}$$

Consider the subset of $\mathfrak{A}(\mathsf{P},\mu)$ with power specifications Q. Observe that the set of rate tuples that this subset achieves is the subset of all simple successive decoding rate tuples dominated by the corner point that corresponds to an adjacent set of power intervals bordering on the channel noise variance. Since this adjacent set is also in $\mathfrak{A}(\mathsf{P},\mu)$, the collection of the dominating rate tuples for each Q for this μ is a dominant subset of rate tuples in $\mathfrak{R}(\mathsf{P},\mu)$. In short, $\mathfrak{R}(\mathsf{P},\mu)$ has a dominant surface.

Since $\mathfrak{D}(\mathsf{P})$ is the simple union of $\mathfrak{R}(\mathsf{P},\mu)$ for all μ, we conclude that it also has a dominant surface. In particular, the dominant surface of $\mathfrak{D}(\mathsf{P})$ is a subset of the union of the dominant surface of $\mathfrak{R}(\mathsf{P},\mu)$. Let function $\mathfrak{D}_s()$ denote the dominant surface of the argument. We have

$$\mathfrak{D}_s(\mathfrak{D}(\mathsf{P})) \subseteq \bigcup_{\forall \mu \in \mathbf{\Pi}(N)} \mathfrak{D}_s(\mathfrak{R}(\mathsf{P},\mu)) \tag{3.91}$$

Since every corner point on the dominant face of the AWGN multi-access achievable rate region

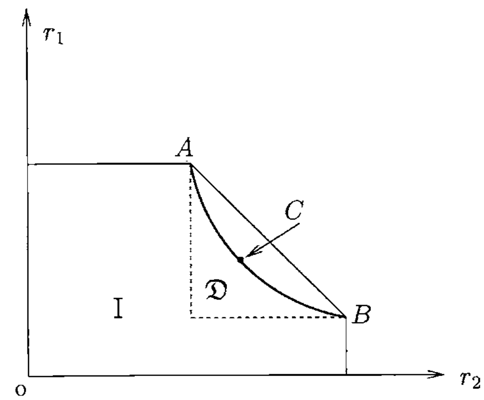


Figure 3-8: The power-reduced successive decoding rate region $\mathfrak{D}(\mathsf{P})$ for a set of 2-users

belongs to a unique $\mathfrak{D}_s(\mathfrak{R}(\mathsf{P},\mu))$, we conclude that the union in (3.91) must be taken over all μ.

Now, the subset of $\mathfrak{D}_s(\mathfrak{R}(\mathsf{P},\mu))$ that is part of $\mathfrak{D}_s(\mathfrak{D}(\mathsf{P}))$ may be identified with the help of the ordering in the r-maximal allocation set. Specifically, a rate tuple R on $\mathfrak{D}_s(\mathfrak{D}(\mathsf{P}))$ that are common to multiple $\mathfrak{D}_s(\mathfrak{R}(\mathsf{P},\mu))$ reflects the existence of multiple decoding orders for the r-maximal allocation set.

In conclusion, the power-reduced successive decoding rate region is dominated by a dominant surface – just like the AWGN multi-access achievable rate region. This dominant surface consists in general of $N!$ sections, each a part of the dominant surface of a distinct decoding order. In particular, a rate tuple on the dominant surface of $\mathfrak{D}(\mathsf{P})$ corresponds to a set of user specifications with an r-maximal allocation set that is adjacent and borders on the given channel's noise variance. Finally, the boundary that marks the transition from one $\mathfrak{D}_s(\mathfrak{R}(\mathsf{P},\mu))$ to another consists of rate tuples R for which multiple decoding orders exist.

Figures 3-8 and 3-9 illustrates this power-reduced successive decoding rate region $\mathfrak{D}$ for the 2- and 3-user cases. In Figure 3-8, section AC of the dominant surface is contributed by the decoding order $\{1,2\}$, and section BC is contributed by the decoding order $\{2,1\}$, i.e. user 2 is decoded last, and user 1 is decoded first. In Figure 3-9, the section of the dominant surface marked by points A, B, C, and D is contributed by the decoding order $\{1,2,3\}$, where the segment BC marks the transition from decoding order $\{1,2,3\}$ to $\{1,3,2\}$, and the segment DC marks the transition from decoding order $\{1,2,3\}$ to $\{2,1,3\}$.

Finally, as a point of curiosity, observe that there is one rate tuple on the dominant face of every power-reduced successive decoding rate region that is common to the contributions of every decoding order, e.g. points C in Figures 3-8 and 3-9. Indeed, this is the rate tuple at which the specifying power intervals have identical upper boundaries. Let this common upper

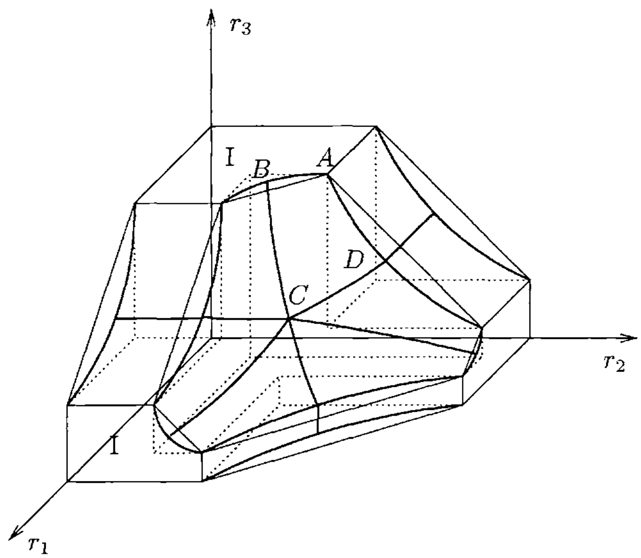


Figure 3-9: The power-reduced successive decoding rate region $\mathfrak{D}(\mathsf{P})$ for a set of 3-users

boundary be at power level b, then

$$\mathcal{U}(\mathsf{R}) = \{(b - p_i, b], i \in [1..N]\} \tag{3.92}$$

We may repeatedly apply the analogous result for the r-maximal allocation set as in Theorem 3.2.7 to conclude that the only condition required for this set of user specifications to be successively decodable is

$$\mathsf{r}((n_o, b]) \geq \sum_{i=1}^{N} \mathsf{r}((b - p_i, b]) \tag{3.93}$$

The optimality of this point dictates that this condition must be satisfied with equality. Multiplying both sides of the resulting equation by 2 and exponentiating, we have

$$\frac{b}{n_o} = \prod_{i=1}^{N} \frac{b}{b - p_i} \tag{3.94}$$

which can be solved as an N^{th} order equation. The positive solution of b that is greater than n_o is the desired solution. Other rate tuples on $\mathsf{B}(\mathfrak{D}(\mathsf{P}))$ may be similarly computed, taking care to order the power intervals by their upper boundaries.

Chapter 4

The Power Diagram and User-Splitting in AWGN Multi-Access Channels

In the previous chapters, we studied achievability and two successive decoding techniques for AWGN multi-access channels from a new perspective. This new perspective is based on a new set of associations of the user and the channel attributes. Under the new associations, achievability questions can be studied directly in terms of the user rate and power specifications. Some such questions we have considered so far include:

1. what is the maximal noise threshold for the set to be achievable?
2. what is the maximal noise threshold for the set to be SSD?
3. and what is the maximal noise threshold for the set to be PR-SD?

To address such questions, we introduced a new framework called the power diagram. This framework makes it possible to place and graphically manipulate users' power intervals, each being an equivalent representation of one user's rate and power constraint, on the same (universal) power axis. For a set of N user specifications, the outcome of our approach was three $O(N \ln N)$ solutions, one to each of the questions posed above.

Finally, when the channel noise is brought back into the picture, the solution to the first question above resulted in a dramatic simplification of the AWGN multi-access achievability condition. The solution to the second question simplified the brute-force approach which requires $N!$ computations. And as far as the author is aware, the solution to the last question is the first known solution for the power-reduced successive decoding region. In addition, the solutions developed for the last two questions also provide, as a side benefit, possible decoding orders in AWGN multi-access channels in which simple successive decoding or power-reduced successive decoding can be used for the given set of user specifications.

In this and the next chapter, we take advantage of this perspective and the power diagram framework, and continue our studies on successive decoding techniques. Specifically, we consider introducing *user-splitting* and *time-sharing* to successive decoding, and address the following questions in the two chapters respectively:

4. what is the maximal noise threshold for a given set of user specifications to be successively decodable with user-splitting?

5. what is the maximal noise threshold for a given set of user specifications to be successively decodable with time-sharing?

As will be seen, the answer to both questions are identical to the maximal noise threshold for the set of user specifications to be achievable. In other words, every rate tuple in the AWGN multi-access achievable rate region may be successively decoded with either user-splitting or time-sharing. Hence, we will only consider combining *user-splitting* and *time-sharing* with simple successive decoding in these two studies.

Also, because the answer to both questions above are identical to the maximal noise threshold for the set of user specifications to be achievable, we will shift the emphasis in these two chapters slightly. In particular, we will focus less on achievability issues and more on splitting algorithms and transmission strategy.

In particular, for a set of N user specifications, we will show that neither combination of techniques requires more than $2N-1$ transmission codebooks for successively decoding any achievable set of user specifications in an AWGN channel. This result, together with the $O(N \ln N)$ solution to AWGN achievability found in Chapter 2, states that the complexity of an N-user AWGN multi-access channel grows no faster than a constant multiple of the combined complexity of N AWGN single-user channels with an ordering. In essence, these results indicate that, at least in theory, an N-user AWGN multi-access channel is no more complex than $2N-1$ single-user channels with an ordering.

In the rest of this chapter, we study combining *user-splitting* and simple successive decoding.

4.1 Introduction

In AWGN multi-access channels, the technique of *user-splitting* divides a user's message and transmission power into several *virtual users* so that each *virtual user* transmits part of the original user's message with a portion of the original user's power, and with its own independently chosen codebook. In this manner, the specifications of each original user is achieved as the simple sum of the specifications of a number of independent *virtual users.*

Definition 4.1.1. A set of users $\mathcal{V}$ is said to be a *splitting-set* of a user T if T is the combined user of $\mathcal{V}$, i.e. $T = \mathsf{T}(\mathcal{V})$. Every member of $\mathcal{V}$ is a *virtual user* of T.

From this definition, a set of users is a *splitting-set* of their combined user. Notice that a given user (with non-zero power and rate) may have a splitting-set that consists of an arbitrary non-negative number of user-splits. In addition, for each $M > 1$, there are many (in fact, uncountably many) splitting-sets consisting of M members. To avoid confusion, we note that a splitting-set as defined above may consist of overlapping power intervals. However, in the rest of this chapter, we will be primarily concerned with splitting-sets consisting of separated power intervals.

Definition 4.1.2. A set of users $\mathcal{V}$ is said to be a *splitting-set* of a second set of users $\mathcal{U}$ if there exists a partition of $\mathcal{V}$, and a one-to-one correspondence between the blocks of the partition and members of $\mathcal{U}$, such that every block is a splitting-set of the corresponding user in $\mathcal{U}$.

Similarly, observe that a given set of N-user specifications may have a splitting-set containing M virtual users for all $M \geq N$. For each $M > N$, there are uncountably many splitting-sets for the same given set.

While it is uncommon[1] to split the user in an AWGN single-user channel into two, this technique has been proven to be potentially useful for multiple user channels. Recall that the dimension of the multi-access achievable rate region depends on the number of users in the channel. Therefore, splitting users would enlarge this dimensionality. However, the combined specifications of a splitting-set is constrained by the specification of the corresponding user. We consider splitting the user in a single-user AWGN channel to illustrate the effect of such constraints on the achievable rate region after the splitting.

Let p and r denote the power constraint and transmission rate in an AWGN channel with noise variance n_o. Then the maximal rate at which the user can reliably transmit in this channel is the channel capacity $\mathsf{C}(p, n_o)$, where

$$\mathsf{C}(p, n_o) = \frac{1}{2} \ln \left(1 + \frac{p}{n_o}\right) \tag{4.1}$$

[1]It is worth noting that, in practice, user-splitting can be desirable in single-user channels for giving the more important parts of the user's message more protection.

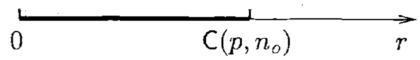


Figure 4-1: The achievable rate region of a single-user AWGN channel

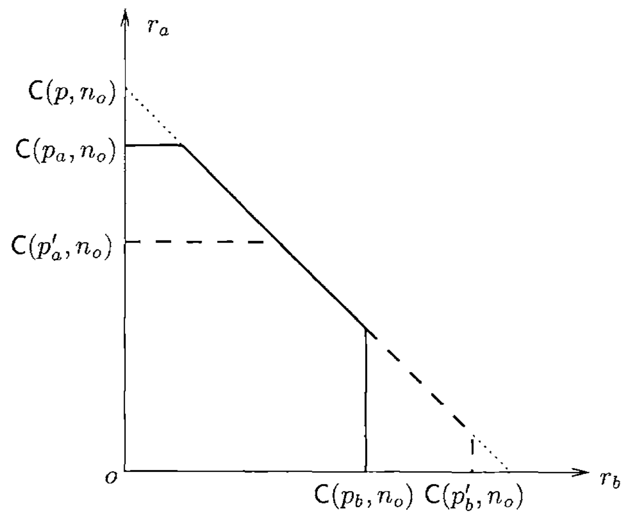


Figure 4-2: The achievable rate region of the AWGN channel after splitting the single user

As illustrated in Figure 4-1, the achievable rate region of this channel is

$$r \leq \mathsf{C}(p, n_o) \tag{4.2}$$

Consider splitting this user into two. In particular, let (p_a, r_a) and (p_b, r_b) respectively denote the specifications of virtual users a and b of the user. The achievable rate region for the two virtual users in the presence of the same channel noise consists of rate tuples (r_a, r_b) satisfying

$$r_a \leq \mathsf{C}(p_a, n_o) \tag{4.3}$$

$$r_b \leq \mathsf{C}(p_2, n_o) \tag{4.4}$$

$$r_a + r_b \leq \mathsf{C}(p_a + p_b, n_o) \tag{4.5}$$

Now, the definition of splitting-set imposes the constraint

$$p_a + p_b = p \tag{4.6}$$

From (4.5)

$$r_a + r_b \leq \mathsf{C}(p, n_o) \tag{4.7}$$

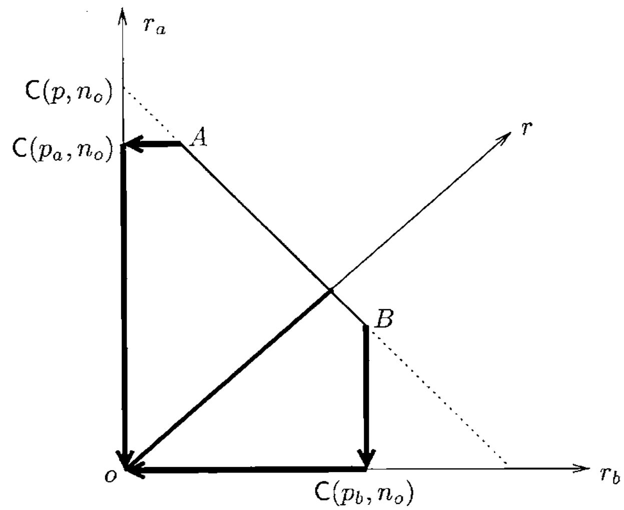


Figure 4-3: Visualizing the effect of splitting the single user into two virtual users

Notice that p_a and p_b are not explicitly involved in the third condition (4.7). In other words, condition (4.5) after the splitting is a mere restatement of condition (4.2) in the original specifications.

From this simple example, we see that splitting users appends new conditions to achievability. In particular, notice that different power splits, i.e. different (p_a, p_b) pairs, only alter the location of the corner points along a fixed line. Figure 4-2 depicts the achievable rate region of the channel with the two different power splits (p_a, p_b) and (p'_a, p'_b). Note that in the extreme cases when either $p_a = 0$ or $p_b = 0$, the two corner points coincide, and we have the original specification with a single user. It is this additional freedom of choosing where to place the corner points that has proven to be useful.

Figure 4-3 offers a visualization of the effect of splitting in this example by super-imposing the achievable rate regions of this channel before and after the splitting. One may perceive that the splitting of the user transformed the interval achievable rate region on the r-axis (before splitting) into the pentagonal achievable rate region between the two rate axis.

Additionally, recall that the corner point A on the dominant face may be successively decoded with decoding order $\{b, a\}$, i.e. with virtual user b decoded first. Similarly, recall that point B may be successively decoded with decoding order $\{a, b\}$. For convenience of visualization, we mark the decoding orders on the achievable rate region for these two corner rate tuples with the linked arrows starting respectively from point A and B.

Historically, Carleial first used this technique to study the achievable rate region of the 2-user interference channel in his thesis [4]. There, he also described successive decoding in the 2-user AWGN multi-access channel. These ideas were also discussed [17] by van der Meulen for decoding the users in the 2-user discrete memoryless multi-access channel. Later, in [5], Cover

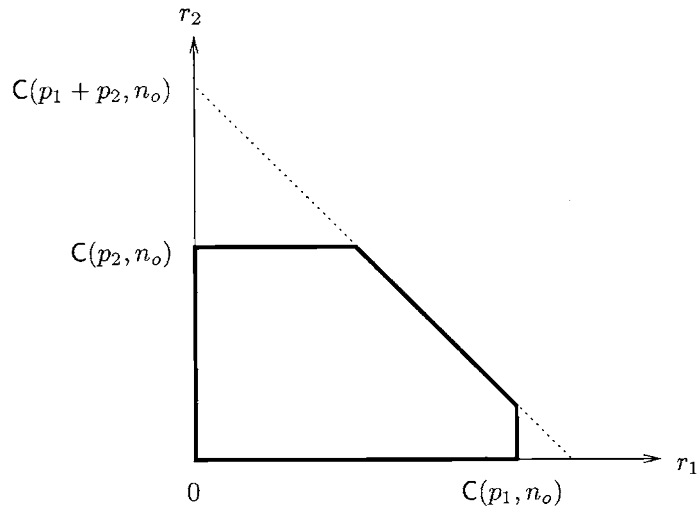


Figure 4-4: The achievable rate region of a two-user AWGN channel

used this same technique to achieve all rate tuples in a achievable rate region of the AWGN *broadcasting* channel, for which Bergmans had established a converse in [2]. Together, these two works established the achievable rate region for the AWGN broadcasting channel.

In more recent years, significant progress in the area of achieving successive decoding for all rate tuples in the AWGN multi-access achievable rate region was made with the *user-splitting* technique. To gain a sense of how any achievable rate tuple may be successively decoded with user-splitting, we incorporate the understandings in achievable rate region for splitting a single user discussed above to consider splitting one user in a two-user AWGN multi-access channel.

Let n_o denote the channel noise variance, and for $i = 1, 2$, let (p_i, r_i) denote the specifications of user i. Recall that the achievable rate region of this channel is a pentagonal region consisting of rate tuples (r_1, r_2) satisfying

$$r_1 \leq \mathsf{C}(p_1, n_o) \tag{4.8}$$

$$r_2 \leq \mathsf{C}(p_2, n_o) \tag{4.9}$$

$$r_1 + r_2 \leq \mathsf{C}(p_1 + p_2, n_o) \tag{4.10}$$

This achievable rate region is illustrated in Figure 4-4.

Let $\mathsf{R} = (r_1, r_2)$ denote a rate tuple on the dominant face of the two-user achievable rate region that is not a corner point. From Chapter 3, R is not SSD[2]. To successively decode this rate tuple using user-splitting, we consider splitting (the power and rate specifications of) user

[2] In fact, R is not successively decodable even with power-reduction and allowing both users to choose arbitrary transmission codebooks.

1 into virtual users a and b. Let (p_a, r_a) and (p_b, r_b) denote the specifications of virtual users a and b respectively. By the definition of virtual user, we have

$$p_a + p_b = p_1 \tag{4.11}$$

$$r_a + r_b = r_1 \tag{4.12}$$

Since R is on the dominant face of the original two-user channel, we have

$$r_1 + r_2 = \mathsf{C}(p_1 + p_2, n_o) \tag{4.13}$$

Combining with (4.11) and (4.12), we have

$$r_a + r_b + r_2 = r_1 + r_2 = \mathsf{C}(p_1 + p_2, n_o) = \mathsf{C}(p_a + p_b + p_2, n_o) \tag{4.14}$$

In other words, the rate tuple after the splitting (r_a, r_b, r_2) must be on the dominant face of the AWGN channel with users a, b, and 2. From discussions in Chapter 3, we know that a rate tuple on the dominant face is successively decodable if and only if it is one of the corner points. Therefore, successive decoding of the rate tuple R with user-splitting is accomplished by finding the appropriate (p_a, r_a) and (p_b, r_b) such that (r_a, r_b, r_2) is a corner point of the achievable rate region with the three users.

To arrive at the desired splitting of user 1, observe that the achievable rate region for the three users a, b, and 2 consists of rate tuples (r_a, r_b, r_2) satisfying

$$r_a \leq \mathsf{C}(p_a, n_o) \tag{4.15}$$

$$r_b \leq \mathsf{C}(p_b, n_o) \tag{4.16}$$

$$r_2 \leq \mathsf{C}(p_2, n_o) \tag{4.17}$$

$$r_a + r_b \leq \mathsf{C}(p_a + p_b, n_o) = \mathsf{C}(p_1, n_o) \tag{4.18}$$

$$r_a + r_2 \leq \mathsf{C}(p_a + p_2, n_o) \tag{4.19}$$

$$r_b + r_2 \leq \mathsf{C}(p_b + p_2, n_o) \tag{4.20}$$

$$r_a + r_b + r_2 \leq \mathsf{C}(p_a + p_b + p_2, n_o) = \mathsf{C}(p_1 + p_2, n_o) \tag{4.21}$$

Figure 4-5 illustrates this achievable rate region.

Now, observe that conditions (4.17), (4.18), and (4.21) are re-statements of the two-user achievability conditions (4.8) through (4.10). As noted in the single user case above, splitting users appends additional conditions to achievability.

Moreover, observe that these conditions are identical for all possible splitting of user 1's

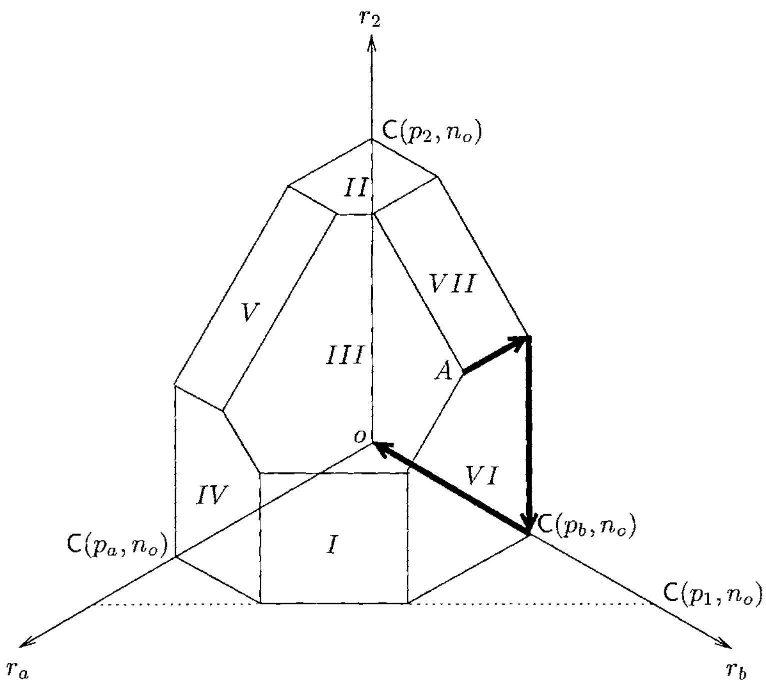


Figure 4-5: Splitting one of the two users into two splits

power into p_a and p_b. Recall that each of these conditions specifies a bounding face of the achievable rate region. We conclude that the planes on which the bounding faces I through III of the achievable rate region are located do not vary with power allocations of p_a and p_b. Note , however, that these bounding faces do vary with the power allocations. In particular, the edges of these bounding faces are determined by the planes on which the remaining four bounding faces are located.

From (4.11), observe that increasing p_a decreases p_b by the same amount. Moreover, notice that increasing p_a simultaneously increases the right sides of conditions (4.15) and (4.19), while decreasing the right sides of (4.16) and (4.20). By the continuity of $\mathsf{C}(\cdot, n_o)$ function, all four terms mentioned above vary continously. These variations corresponds to continuously moving bounding faces IV and V away from the origin, and continuously moving bounding faces VI and VII towards the origin (while the norm of all four planes are kept constant).

Finally, recall that the rate tuple B is a corner point that corresponds to successive decoding order $\{a, 2, b\}$. In particular, B satisfies conditions (4.16), (4.20) and (4.21) with equality, i.e.

$$r_b = \mathsf{C}(p_b, n_o) \tag{4.22}$$

$$r_b + r_2 = \mathsf{C}(p_b + p_2, n_o) \tag{4.23}$$

$$r_a + r_b + r_2 = \mathsf{C}(p_1 + p_2, n_o) \tag{4.24}$$

Substituting (4.22) into (4.23),

$$r_2 = \mathsf{C}(p_b + p_2, n_o) - \mathsf{C}(p_b, n_o) = \mathsf{C}(p_2, n_o + p_b) \tag{4.25}$$

Observe that the r_2 component of B increases continuously from $\mathsf{C}(p_2, n_o + p_1)$ to $\mathsf{C}(p_2, n_o)$ as p_b decreases from p_1 to 0. Hence, there exists a $p_b^* \in [0, p_1]$ such that the r_2 component of B is identical to r_2, i.e.

$$\mathsf{C}(p_2, n_o + p_b^*) = r_2 \tag{4.26}$$

Combining with (4.22), we have the desire specification of virtual user b as $(p_b^*, \mathsf{C}(p_b^*, n_o))$. The desired specification of virtual users a can be obtained using (4.11) and (4.12).

This procedure of obtaining the desired specification of the virtual users of user 1 may be perceived from super-imposing Figures 4-4 and 4-5 as in the single-user case above. In particular, we match the r_2 axis, and pick an arbitrary direction between the r_a and the r_b axis to be the r_1 axis as in Figure 4-6. For convenience, we mark the borders of the achievable rate region of the original two users in this channel with dashed lines. In particular, the desired rate tuple B is located by varying p_b until B is on the same perpendicular plane to the r_2 axis as R.

Incidentally, the above procedure also shows that only one user needs to be split in the two-user case, and by the arbitrary choice of users 1 and 2, we see that either user may be chosen to be un-split.

Extending the above procedure of splitting two users to N arbitrary users proves to be non-trivial. One way to perceive the reason is through the difficulty in directly visualizing the constructions with achievable rate regions in greater than 3 dimensional space (with more than 3 virtual users). At a more fundamental level, the difficulty resides in understanding the effect that a decision on how to split (and whether to split) a user has on the decision on how (and whether) to split the other users.

In 1996, Rimoldi and Urbanke [12] presented an approach to understanding such effects using a construct that is a pre-cursor to the overlapping set of power intervals introduced in Chapter 2. They showed that it is possible to facilitate simple successive decoding with *user-splitting* to every rate tuple on the dominant face of the N-user AWGN multi-access achievable rate region. In particular, they showed that it is possible to construct a SSD splitting-set such that no user in the original specification is required to be split into more than 2 virtual users, and such that at least one user need not be split. They further argued that, since any rate tuple in the interior of the AWGN multi-access achievable rate region is component-wise dominated by at least one rate tuple on the dominant face, then every rate tuple in the AWGN multi-access

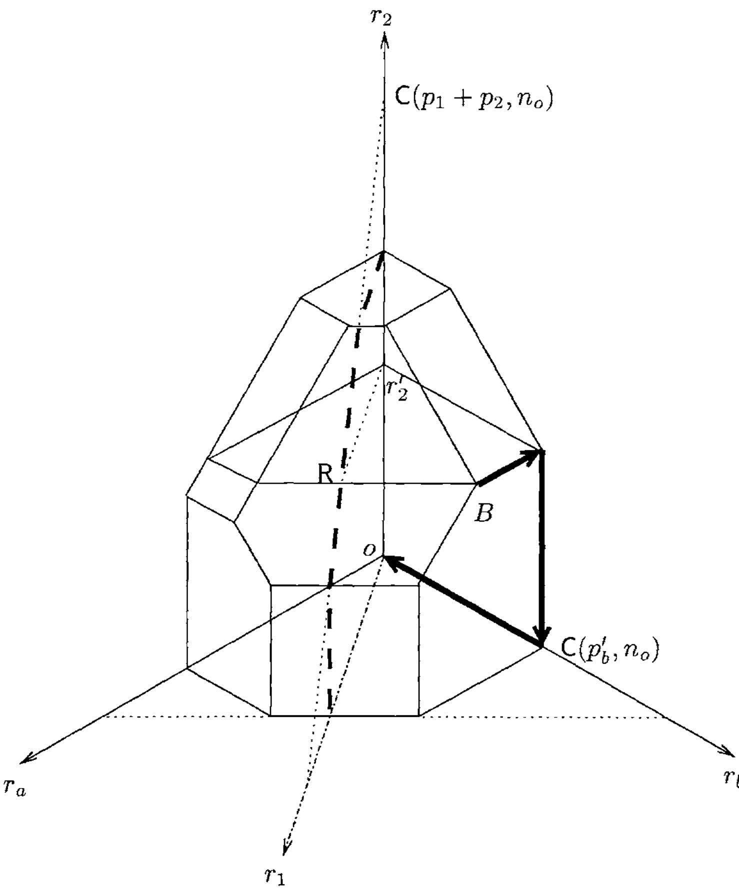


Figure 4-6: Splitting one of the two users into two splits to achieve simple successive decoding

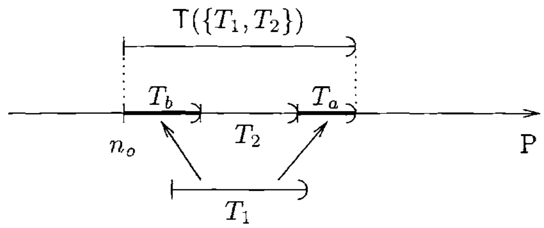


Figure 4-7: Splitting one of the two users into two to achieve simple successive decoding in the power diagram

achievable rate region is SSD with *user-splitting.*

From the previous chapters, we have seen that the power diagram can be used to consider sets of N arbitrary users on the same power axis. We will now see that the power diagram can be used to construct splitting-sets for general N-user AWGN multi-access channels. We illustrate this approach by interpreting the two-user example discussed above using the power diagram.

For $i = 1, 2$, let T_i denote the power interval of user i's specifications (p_i, r_i). Since R is on the dominant face before splitting, from Chapter 2, we know that T_1 and T_2 overlap with their combined power interval bordering on the channel noise intensity n_o.

Let T_a and T_b denote the power intervals of the specifications of virtual users a and b. In particular, let

$$\mathsf{p}(T_a) = p_1 - p_b^* \tag{4.27}$$

$$\mathsf{p}(T_b) = p_b^* \tag{4.28}$$

$$\mathsf{r}(T_a) = r_1 - \mathsf{r}(T_b) \tag{4.29}$$

$$\mathsf{r}(T_b) = \mathsf{C}(p_b^*, n_o) \tag{4.30}$$

From the above, $\{T_a, T_b, T_2\}$ corresponds to a corner point with decoding order $\{a, 2, b\}$ after splitting. From Chapter 2, $\{T_a, T_b, T_2\}$ is a set of adjacent power intervals with the lower boundary of T_b at the channel noise intensity n_o.

Since the combined user of T_1 and T_2 borders on the channel noise intensity n_o, and is identical to the combined user of $\{T_a, T_b, T_2\}$, this splitting result may be seen as simply *dividing* the combined power interval of the original user specifications into a set of adjacent power intervals which contains T_2. Figure 4-7 presents this splitting-set construction.

Recall, by definition, that the extent of a set of adjacent power intervals $\mathcal{U}$ is the union of the set. For convenience of discussion, we will call such a $\mathcal{U}$ the *division* of $\mathsf{ext}(\mathcal{U})$.

In this 2-user example, $\{T_a, T_b, T_2\}$ is a division of $\mathsf{T}(\{T_1, T_2\})$. We may alternatively use results from Section 3.2 to directly argue that this division is SSD in the given channel as follows. First, by definition, we know that the p-maximal allocation set of a set of disjoint power intervals is itself. Combining with Theorem 3.2.4, this division is SSD in any channel with noise intensity below $\eta(T_b)$. Note that this argument can be used to show that a disjoint set of power intervals is SSD in any AWGN channel in which it is achievable. Finally, we observe that $\eta(T_b) = \eta(\mathsf{T}(\{T_1, T_2\}))$ and arrive at the desired conclusion.

Observe that successive decoding with user-splitting in this case is accomplished via constructing a division of the combined user of the two user power intervals T_1 and T_2 that is also a splitting-set of the two users. This approach to successive decoding with user-splitting turns out to be generalizable for N arbitrary users.

Definition 4.1.3. Let $\mathcal{V} = \{v_i, i \in [1, \dots, K]\}$ be a set of disjoint intervals, such that for each $i \in [1, \dots, K]$, v_i is the extent of a set of adjacent intervals $\mathcal{W}_i$. Then $\cup_{i=1}^{K} \mathcal{W}_i$ is said to be a *division set* (or a division) of $\mathcal{V}$.

Figure 4-8 illustrates this definition with $\mathcal{W}_i = \{w_{i,j}, j \in [1, \dots, K_i]\}$.

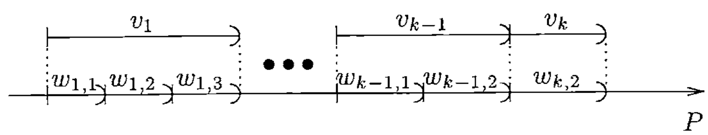


Figure 4-8: A division set $\mathcal{W}$ of a set of disjoint power intervals $\mathcal{V}$

We will call the operation of constructing a division set a *dividing* operation.

Although Rimoldi and Urbanke developed their user-splitting construction in [12] without the convenience of graphical constructions offered by the power diagram framework, their construction may be understood more easily using the power diagram framework. From Chapter 2, we know that a rate tuple on the dominant face of an AWGN multi-access achievable rate region corresponds to a set of user power intervals $\mathcal{U}$ that has the overlapping property with $\eta(\mathsf{T}(\mathcal{U})) = n_o$, where n_o is the channel noise intensity. Finally, using the concepts developed above, one may regard their construction as one that divides $\mathsf{T}(\mathcal{U})$ into a splitting-set of $\mathcal{U}$ such that the splitting-set of no member of $\mathcal{U}$ contains more than 2 virtual users.

In [16], I presented the power diagram framework. Then, using this framework, a class of graphical constructions was developed, each member of which divides the irreducible equivalent of a given set of N-user specifications into a splitting-set. From this point of view, one may regard these constructions as extensions of [12] for user-splitting on the dominant face to arbitrary achievable set of user specifications, where users in each constructing block are split within the corresponding member of the irreducible equivalent.

The advantage of such extensions in [16] may be readily seen with rate tuple on a bounding surface of the channel's achievable rate region other than the dominant face. In an AWGN multi-access channel with noise intensity n_o, let a set of user specifications $\mathcal{U}$ corresponds to a rate tuple on a bounding surface of the channel's achievable rate region other than the dominant face. From Chapter 2, recall the irreducible equivalent of such a $\mathcal{U}$ consists of more than one power intervals, with the lowest member, $[\mathsf{irr}(\mathcal{U})]_1$, bordering on the channel noise intensity n_o. Additionally, we also know that there is a gap between $[\mathsf{irr}(\mathcal{U})]_1$ and the next lowest member $[\mathsf{irr}(\mathcal{U})]_2$. In other words, there exists additional signal-to-noise margins for the subset of users in $\mathsf{B}_2(\mathcal{U})$ (and other constructing blocks of $\mathcal{U}$, if there is any). By performing user-splitting for users not in the first constructing block above $[\mathsf{irr}(\mathcal{U})]_1$, such signal-to-noise margins are preserved for these users.

Additionally, I showed in [16] that there are multiple constructions in this class that lead to a total of no more than $2N-1$ virtual users. In particular, the construction established in [12] is optimal under certain system requirements, while others may be more desirable under other requirements.

In this chapter, we fully develop the ideas in [16] from a difference perspective which contributes towards additional generality. To perceive the desirability of such additional generalities, note that all user-splitting constructions presented in both [12] and [16] are divisions of the combined user of an overlapping set. However, neither work addresses whether there exist other interesting and desirable user-splitting constructions.

This question is partially addressed by the results in Section 3.2. In particular, we may argue that constructing splitting-sets consisting of disjoint virtual users is the sensible thing to do. To see this, first recall, from Theorem 3.2.4, that a set of user specifications is SSD in a given channel if and only if its p-maximal allocation set is achievable in the channel. The p-maximal allocation set consists of disjoint power intervals which have a unique correspondence to users in the original set of user specifications. Thus any SSD splitting-set of $\mathcal{U}$ must have a p-maximal allocation set that is achievable (and hence SSD) in the given channel.

Now, let T' denote the power interval in the p-maximal allocation set for a given user with power interval T. By definition of the p-maximal allocation set, $\mathsf{p}(T') = \mathsf{p}(T)$, and the upper boundary of T' is lower than that of T. By the monotonicity of the $\mathsf{r}(\cdot)$ function, $\mathsf{r}(T') \geq \mathsf{r}(T)$. In other words, disjoint splitting-sets provide better signal-to-noise margin for decoding the users compared to their non-disjoint counterparts. From this point-of-view, it makes sense to consider only the disjoint splitting-sets.

The remaining question is

are there interesting and desirable user-splitting constructions that are not divisions of the irreducible equivalent of the given set of user specifications?

The answer to this question turns out to be affirmative. We illustrate this point with a simple example consisting of two overlapping power intervals T_1 and T_2.

From the above discussions, we know that there exists a splitting-set $\{T_a, T_b\}$ of T_1, such that $\{T_a, T_b, T_2\}$ is a division of $\mathsf{T}(\{T_1, T_2\})$. In particular, we may choose the splitting-set of T_1 as in Figure 4-7, i.e.

$$T_b < T_2 < T_a \tag{4.31}$$

Consider holding $\mathsf{r}(T_b)$ fixed while lowering the boundaries of T_b to arrive at T_b'. By the monotonicity of $\mathsf{r}(\cdot)$ function, we have

$$\mathsf{p}(T_b') < \mathsf{p}(T_b) \tag{4.32}$$

Let T_a' be such that $\{T_a', T_b'\}$ is a splitting-set of T_1. By definition of splitting-set, we have

$$\mathsf{r}(T_a') = \mathsf{r}(T_1) - \mathsf{r}(T_b') = \mathsf{r}(T_1) - \mathsf{r}(T_b) = \mathsf{r}(T_a) \tag{4.33}$$

$$\mathsf{p}(T_a') = \mathsf{p}(T_1) - \mathsf{p}(T_b') \geq \mathsf{p}(T_1) - \mathsf{p}(T_b) = \mathsf{p}(T_a) \tag{4.34}$$

By the monotonicity of $\mathsf{r}(\cdot)$ function, we have

$$\eta(T_a) < T_a' \tag{4.35}$$

We have thus arrived at an alternate splitting-set of $\{T_1, T_2\}$ which consists of the disjoint power intervals $\{T_a', T_b', T_2\}$. As seen in Figure 4-9, this new splitting-set is not a division of $\mathsf{T}(\{T_1, T_2\})$.

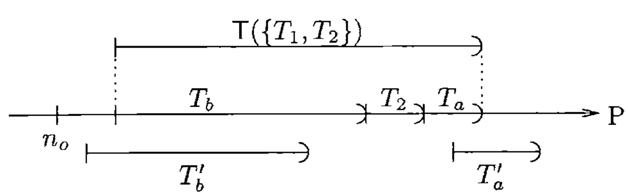


Figure 4-9: Illustrating the case when the splitting-set $\{T_a', T_b', T_2\}$ is more desirable than a division of $\mathsf{T}(\{T_1, T_2\})$

Suppose the channel noise variance n_o is lower than $\eta(T_b')$. From discussions in Section 3.2, we know that $\{T_a', T_b', T_2\}$ is SSD if achievable. Hence this splitting-set may be used to facilitate successive decoding to the original set of user specifications $\{T_1, T_2\}$ with user-splitting.

Observe that by (4.32), better signal-to-noise ratio is made available for decoding user 2 by successively decoding $\{T'_a, T'_b, T_2\}$ instead of $\{T_a, T_b, T_2\}$. This can be very desirable, particularly when the power constraint of user 1 is much much larger than that of user 2, and the power specification of T_a constructed consists of a very small portion of $\mathsf{p}(T_2)$, as illustrated in Figure 4-9.

To generalize to sets of arbitrary N user specifications $\mathcal{U}$, recall that $\mathsf{irr}(\mathcal{U})$ may consist of multiple separated power intervals. Suppose $\mathsf{B}_j(\mathcal{U}) = \{T_1, T_2\}$ for some $j > 1$. Recall that in such cases, the maximal noise threshold for the set to be achievable does not depend on $[\mathsf{irr}(\mathcal{U})]_j = \mathsf{T}(\{T_1, T_2\})$. Then using a splitting-set of $\mathsf{B}_j(\mathcal{U})$ in the form of $\{T'_a, T'_b, T_2\}$ need not affect the achievability of the splitting-set of $\mathcal{U}$. It is therefore clear that disjoint splitting-sets that are not divisions of the irreducible equivalent can be more desirable.

As will be seen shortly, our approach in this chapter includes all such disjoint splitting-sets. In particular, we pose and answer the generic question

> given a set of user specifications $\mathcal{U}$ and a set of separated power intervals $\mathcal{V}$, does there exist a division of $\mathcal{V}$ that is also a splitting-set of $\mathcal{U}$?

Definition 4.1.4. A set of separated power intervals $\mathcal{V}$ is said to be *suitable* to a set of user specifications $\mathcal{U}$ if there exists a division of $\mathcal{V}$ that is also a splitting-set of $\mathcal{U}$.

Using this definition, the answer to the above question is seen as the conditions for suitability of $\mathcal{V}$ to $\mathcal{U}$. For this reason, we will refer to such an answer as the *suitability conditions.* Observe that in the two-user example above, $\mathsf{T}(\{T_1, T_2\})$ is suitable to $\{T_1, T_2\}$.

The rest of this chapter is organized as follows. In Section 4.2, we extend the two-user case discussed above into a simple user-splitting construction algorithm that is based on the irreducible equivalent construction. Then, in Section 4.3, we use properties of the constructions resulting from this algorithm to develop a set of simple suitability conditions. Finally, in Section 4.4, we present a class of algorithms based on the suitability conditions developed and the *conservation principle* (to be defined). We will show that for a given set of user specifications, the set of user-splitting constructions resulting from these algorithms subsumes all previously known such constructions. In fact, we will show that this set includes all possible disjoint splitting-set constructions.

Before leaving this section, we note that the *user-splitting* technique was also successfully applied to the general discrete memoryless multi-access channels to facilitate successive decoding by Grant et al [9]. They showed that no more than $2N - 1$ splits are required in these channels.

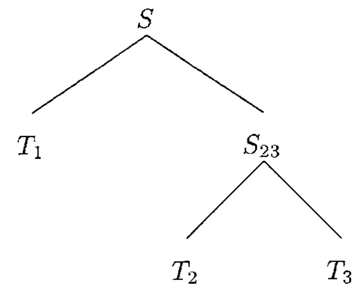


Figure 4-10: The overlapping tree for $\{T_1, T_2, T_3\}$

4.2 The Simple Splitting Algorithm

In this section, we extend the two-user splitting construction discussed in the previous section into a simple user-splitting algorithm.

Recall in the two overlapping users example above (Figure 4-7), the combined user $\mathsf{T}(\{T_1, T_2\})$ is divided into an adjacent set of three power intervals $\{T_a, T_b, T_2\}$. Note that this division contains T_2. Recall when T_2 is excluded from the division, the remaining power intervals T_a and T_b form a splitting-set of the other user T_1. Note that among all possible divisions of $\mathsf{T}(\{T_1, T_2\})$ that contain T_2, the division $\{T_a, T_b, T_2\}$ consists of the least number of power intervals[3], i.e. has the least cardinality.

Finally, note that in the resulting splitting-set, T_1 is split into two virtual users, and T_2 is not split. Clearly, the roles of the two users may be reversed in the construction so that T_1 is not split.

To generalize this construction, first consider the case of three overlapping users $\{T_1, T_2, T_3\}$. By Theorem 2.3.5, these three power intervals must be the leaves of one overlapping tree. Without the loss of generality, let this overlapping tree be as in Figure 4-10.

Note that T_1 and $S_{23} = \mathsf{T}(\{T_2, T_3\})$ overlap. Hence we may obtain a splitting-set for the three users by applying the construction in the two-user case above twice, once to divide $S = \mathsf{T}(\{T_1, T_2, T_3\})$ into a splitting-set of $\{T_1, S_{23}\}$, and a second time to divide S_{23} into a splitting-set of $\{T_2, T_3\}$. These two construction steps are illustrated in Figure 4-11.

Notice that in the resulting splitting-set, T_1 and T_2 are each split into two virtual users, and T_3 is not split. Clearly, one may reverse the roles of T_2 and T_3 in this construction so that T_2 is not split. Also observe that if T_1 overlaps with T_2 (or T_3), we may exchange the role of T_1 and that of T_3 (or T_2), so that T_1 is not split in the resulting division of S. However, if T_1 does not overlap with either T_2 or T_3, can we still devise a division of S (into a splitting-set of the three users) such that T_1 is not split?

[3] Observe that either T_a or T_b may be empty in the case where $\{T_1, T_2\}$ is adjacent.

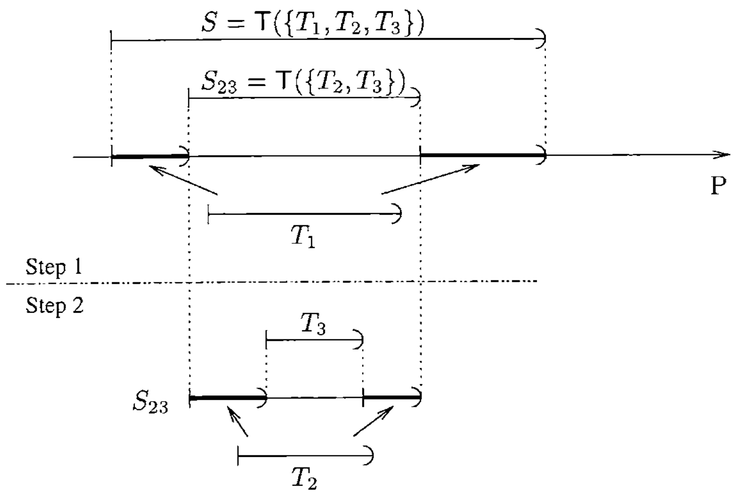


Figure 4-11: One division of the combined user of three overlapping power intervals into their splitting-set

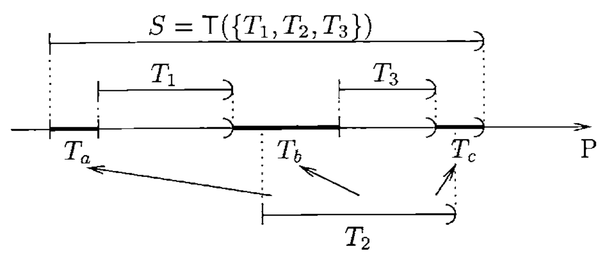


Figure 4-12: A second division of the combined user of three overlapping power intervals into their splitting-set when T_1 does not overlap with T_2 or T_3, and T_1 is not split

The answer to this question is affirmative. Figure 4-12 illustrates such a splitting-set.

In particular, the division of S illustrated in Figure 4-12 is the one with the least number of power intervals that contains both T_1 and T_3. Let this division be $\{T_a, T_b, T_c, T_1, T_3\}$. To show it is the desired splitting-set of the three power intervals, we need to prove that $\{T_a, T_b, T_c\}$ is a splitting-set of T_2.

To see this, observe that since S is the combined user of the three power intervals, the combined power and rate specifications of its division must be identical to the those of the three users. Hence the combined power and rate specifications of $\{T_a, T_b, T_c\}$ must be identical to the power and rate specifications of T_2, which, by definition, states that the division excluding T_1 and T_3 is a splitting-set of T_2.

Clearly, the roles of T_2 and T_3 may be reversed in this construction. Observe that in the resulting division, T_2 is split into three virtual users, while neither T_1 nor T_3 is split.

Using the above constructions on a set of 3 overlapping power intervals, we see that, while any one particular user may be designated to be not split, we were unable to simultaneously satisfy the requirement that no user is split into more than 2 virtual users. It turns out that this second requirement is less straight-forward to fulfill, and we will leave it for later sections.

In the rest of this section, we will generalize the constructions in the above to a simple algorithm. This algorithm divides the irreducible equivalent of an arbitrary set of user specifications $\mathcal{U}$ into a splitting-set of $\mathcal{U}$ that satisfies the condition that any one user in $\mathcal{U}$ may be chosen not to be split. This is accomplished by appropriately generalizing two ideas embedded in the examples above.

First, notice that in both the two- and the three-user cases discussed above, we have constructed divisions with the least number of power intervals which contain some given power intervals in the original set of user specifications. This is the first idea that we generalize.

First, we define the following notation for convenience.

Definition 4.2.1. Let sets $\mathcal{V}$ and $\mathcal{W}$ each consist of disjoint power intervals. Then, we say that $\mathcal{W}$ *precedes* $\mathcal{V}$ *(or* $\mathcal{V}$ *succeeds* $\mathcal{W}$*)* if every member power interval in $\mathcal{W}$ is contained in a member power interval in $\mathcal{V}$, denote it by $\mathcal{W} \prec \mathcal{V}$ (or $\mathcal{V} \succ \mathcal{W}$).

Consider the following definition.

Definition 4.2.2. Let $\mathcal{V}$ denote a set of separated power intervals and $\mathcal{W}$ denote a set of disjoint power intervals such that every member of $\mathcal{W}$ is contained in a member of $\mathcal{V}$, i.e. $\mathcal{W} \prec \mathcal{V}$. Let $\mathcal{V}'$ be a division of $\mathcal{V}$ consisting of the least number of power intervals such that each power interval in $\mathcal{W}$ is a power interval in $\mathcal{V}'$, i.e. $\mathcal{W} \subseteq \mathcal{V}'$. Then $\mathcal{V}' \setminus \mathcal{W}$ is said to be the *complementing set* of $\mathcal{W}$ in $\mathcal{V}$, denote it by $\mathsf{cmpl}(\mathcal{V}, \mathcal{W})$

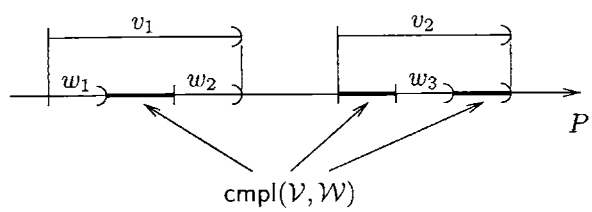


Figure 4-13: The complementing set of $\mathcal{W} = \{w_1, w_2, w_3\}$ in $\mathcal{V} = \{v_1, v_2\}$, $\mathsf{cmpl}(\mathcal{V}, \mathcal{W})$

Observe that in the two overlapping power interval case above, $\{T_a, T_b\}$ is the complementing set of $\{T_2\}$ in $\{\mathsf{T}(\{T_1, T_2\})\}$.

In the following, if $\mathcal{V} = \{S\}$ is a singleton set, we may abbreviate the notation $\mathsf{cmpl}\,(\mathcal{V}, \mathcal{W})$ as $\mathsf{cmpl}(S, \mathcal{W})$ for convenience. Similarly, if $\mathcal{W} = \{T\}$, we may use $\mathsf{cmpl}(\mathcal{V}, T)$ in place of $\mathsf{cmpl}(\mathcal{V}, \mathcal{W})$.

The second idea starts from noticing that, in the case of two overlapping power intervals, we showed that $\mathsf{cmpl}(\mathsf{T}(\{T_1, T_2\}), T_2) = \{T_a, T_b\}$ is a splitting-set of T_1. Similarly, in the three overlapping power interval cases above, $\mathsf{cmpl}(S, S_{23})$ and $\mathsf{cmpl}(S_{23}, T_3)$ in Figure 4-11

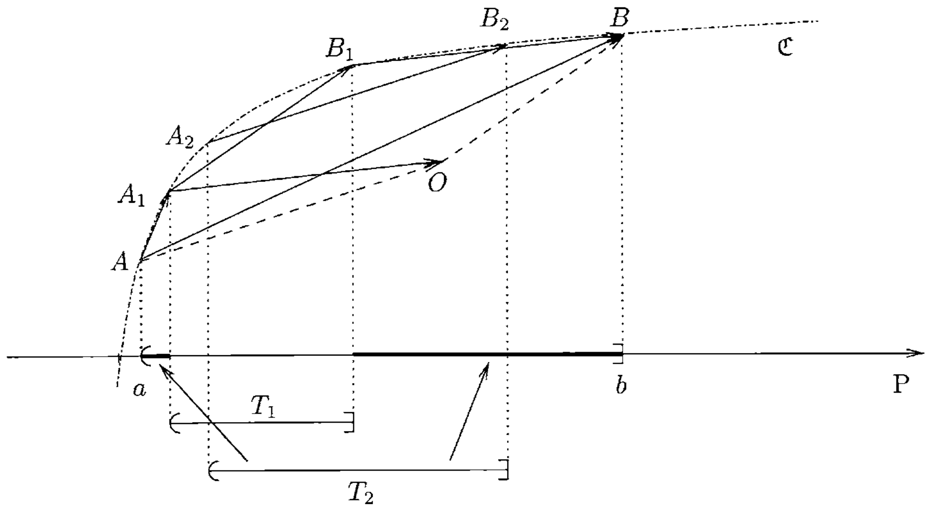


Figure 4-14: Proving the two overlapping users case for Theorem 4.2.1 using the vector representation of power intervals

were respectively shown to be the splitting-sets of T_1 and T_2. In addition, we showed that $\mathsf{cmpl}(S, \{T_1, T_3\})$ in Figure 4-12 is a splitting-set of T_2. We generalize this below.

Theorem 4.2.1. *Let T_1 and T_2 be two overlapping*[4] *power intervals, $\mathcal{V} = \{v_j, j \in [1, \dots, J]\}$ be a separated splitting-set of $\mathsf{T}(\{T_1, T_2\})$, and $\mathcal{W} = \{w_k, k \in [1, \dots, K]\}$ be a disjoint splitting-set of T_1. Suppose $\mathcal{W} \prec \mathcal{V}$. Then $\mathsf{cmpl}(\mathcal{V}, \mathcal{W})$ is a splitting-set of T_2.*

Remark. In the simplest case, $\mathcal{V}$ is a singleton set consisting of $\mathsf{T}(\{T_1, T_2\})$, and $\mathcal{W}$ is a singleton set consisting of T_1. The complementing set of $\mathcal{W}$ in $\mathcal{V}$, $\mathsf{cmpl}(\mathcal{V}, \mathcal{W})$, was shown in the introduction of this chapter to be a splitting-set of T_2.

Figure 4-14 presents an alternative visual proof for this simple case using the vector representation of power intervals introduced in Section 2.1. The ideas embedded in this visual proof is at the center of the proof of this theorem in general. For this reason, we detail this visual proof below.

Let $\{\mathsf{T}(\{T_1, T_2\})\} = (a, b]$, and $T_1 = (a_1, b_1]$, we have $\mathsf{cmpl}(\mathcal{W}, \mathcal{V}) = \{(a, a_1], (b_1, b]\}$. Construct point O using parallelograms such that $\overrightarrow{OB} = \overrightarrow{A_1B_1}$, and $\overrightarrow{AO} = \overrightarrow{A_2B_2}$. The desired conclusion follows from the fact that

$$\overrightarrow{A_2B_2} = \overrightarrow{AO} = \overrightarrow{AA_1} + \overrightarrow{A_1O} = \overrightarrow{AA_1} + \overrightarrow{B_1B} \tag{4.36}$$

Proof. By definition of the complementing set, $\mathsf{cmpl}(\mathcal{V}, \mathcal{W}) \cup \mathcal{W}$ is a division of $\mathcal{V}$. By the

[4] *T_1 and T_2 need not be overlapping for the result of this theorem to be true.*

definition of division, we have

$$\mathsf{T}(\mathcal{V}) = \mathsf{T}(\mathsf{cmpl}(\mathcal{V},\mathcal{W}) \cup \mathcal{V}) \tag{4.37}$$

From the definition of combined user, we have

$$\mathsf{p}(\mathsf{T}(\mathcal{V})) = \mathsf{p}(\mathsf{T}(\mathsf{cmpl}(\mathcal{V},\mathcal{W}) \cup \mathcal{W})) = \mathsf{p}(\mathsf{T}(\mathsf{cmpl}(\mathcal{V},\mathcal{W}))) + \mathsf{p}(\mathsf{T}(\mathcal{W})) \tag{4.38}$$

$$\mathsf{r}(\mathsf{T}(\mathcal{V})) = \mathsf{r}(\mathsf{T}(\mathsf{cmpl}(\mathcal{V},\mathcal{W}) \cup \mathcal{W})) = \mathsf{r}(\mathsf{T}(\mathsf{cmpl}(\mathcal{V},\mathcal{W}))) + \mathsf{r}(\mathsf{T}(\mathcal{W})) \tag{4.39}$$

By assumption, $\mathsf{T}(\mathcal{V}) = \mathsf{T}(\{T_1, T_2\})$, and $T_1 = \mathsf{T}(\mathcal{W})$. Substituting into the equations above, we have

$$\mathsf{p}(T_1) + \mathsf{p}(T_2) = \mathsf{p}(\mathsf{T}(\mathcal{V})) = \mathsf{p}(\mathsf{T}(\mathsf{cmpl}(\mathcal{V},\mathcal{W}))) + \mathsf{p}(\mathsf{T}(\mathcal{W})) = \mathsf{p}(\mathsf{T}(\mathsf{cmpl}(\mathcal{V},\mathcal{W}))) + \mathsf{p}(T_1) \tag{4.40}$$

$$\mathsf{r}(T_1) + \mathsf{r}(T_2) = \mathsf{r}(\mathsf{T}(\mathcal{V})) = \mathsf{r}(\mathsf{T}(\mathsf{cmpl}(\mathcal{V},\mathcal{W}))) + \mathsf{r}(\mathsf{T}(\mathcal{W})) = \mathsf{r}(\mathsf{T}(\mathsf{cmpl}(\mathcal{V},\mathcal{W}))) + \mathsf{r}(T_1) \tag{4.41}$$

Subtracting $\mathsf{p}(T_1)$ from both sides of (4.40), and subtracting $\mathsf{p}(T_1)$ from both sides of (4.40), we have

$$\mathsf{p}(T_2) = \mathsf{p}(\mathsf{T}(\mathsf{cmpl}(\mathcal{V},\mathcal{W}))) \tag{4.42}$$

$$\mathsf{r}(T_2) = \mathsf{r}(\mathsf{T}(\mathsf{cmpl}(\mathcal{V},\mathcal{W}))) \tag{4.43}$$

And the desired conclusion follows from the definition of splitting-set. □

Corollary 4.2.2. *Let $\mathcal{V}$ be a set of disjoint power intervals, and T a power interval. Then $\mathsf{cmpl}(\mathsf{irr}(\mathcal{V} \cup \{T\}), \mathcal{V})$ is a splitting-set of T.*

Proof. By definition, $\mathcal{V}$ is a splitting-set of $\mathsf{T}(\mathcal{V})$.

By definition of irreducible equivalent, $\mathsf{irr}(\mathcal{V} \cup \{T\})$ consists of separated power intervals, and

$$\mathsf{T}(\mathsf{irr}(\mathcal{V} \cup \{T\})) = \mathsf{T}(\mathcal{V} \cup \{T\}) = \mathsf{T}(\{\mathsf{T}(\mathcal{V}), T\}) \tag{4.44}$$

Finally, by the definition of irreducible equivalent and the overlapping property, we also have every member of $\mathcal{V}$ is contained in a member of $\mathsf{irr}(\mathcal{V} \cup \{T\})$. The corollary follows. □

Since an irreducible equivalent consists of separated power intervals, we may repeatedly apply the result of this corollary to divide the irreducible equivalent of a set of N arbitrary power intervals into its splitting-set as in the following algorithm.

Algorithm 4.2.3 (The simple Splitting Algorithm).
Let $\mathcal{U} = \{u_i, i \in [1, \dots, N]\}$ be a set of N power intervals. Initializing $\mathcal{W}_0$ to be $\emptyset$. For each i from 1 to N, construct the following:

Step 1: construct $\mathcal{W}_i = \mathsf{irr}(\mathcal{W}_{i-1} \cup \{T_i\})$

Step 2: construct $\mathcal{V}_i = \mathsf{cmpl}(\mathcal{W}_i, \mathcal{W}_{i-1})$

Construct $\mathcal{V}$ to be the union of $\mathcal{V}_i$, i.e. $\mathcal{V} = \cup_{i=1}^{N} \mathcal{V}_i$.

By Corollary 4.2.2, $\mathcal{V}_i$ is a splitting-set of T_i for each $i \in [1, \dots, N]$. We therefore have the following.

Theorem 4.2.4. *$\mathcal{V}$ is a division of $\mathsf{irr}(\mathcal{U})$ and a splitting-set of $\mathcal{U}$.*

In other words, given a set of power intervals, there exists a division of its irreducible equivalent that is its splitting-set. Hence,

Corollary 4.2.5. *The irreducible equivalent of a set of power intervals is suitable to the set.*

Recall that by Corollary 2.3.9, the maximal noise threshold for $\mathcal{U}$ to be achievable is $\eta([\mathsf{irr}\,(\mathcal{U})]_1)$. We therefore have the following.

Corollary 4.2.6. *The maximal noise threshold for a given set of user specifications to be SSD with user-splitting is the maximal noise threshold for the set to be achievable.*

We make a few observations regarding the above construction algorithm.

First, observe that the two-user example in Figure 4-7 is the outcome of this algorithm with ordering $\{2, 1\}$; and the splitting-set illustrated in Figures 4-11 and 4-12 are the outcome of the algorithm with ordering $\{3, 2, 1\}$ and $\{3, 1, 2\}$ respectively. Notice that the construction in Figure 4-12 can also be arrived at via ordering $\{1, 3, 2\}$.

Second, notice that this construction algorithm succeeds with an arbitrary ordering of members of $\mathcal{U}$ into $\{u_i, i \in [1, \dots, N]\}$. Therefore, for a set of N users, this algorithm can not produce more than $N!$ possible splitting-sets. In addition, note that for every such ordering, $\mathcal{V}_1 = \{u_1\}$ is always a singleton set. Hence there is always one user that is not split.

Third, we note that $\mathcal{V}$ contains no more than $2N - 1$ power intervals. This fact may be seen a number of ways.

One way is through counting the number of power intervals in $\mathcal{V}_i = \mathsf{cmpl}(\mathcal{W}_i, \mathcal{W}_{i-1})$ for a given size of $\mathcal{W}_{i-1}$. Note that $\mathcal{W}_i$ is the irreducible equivalent of $\{u_j, j \in [1, \dots, i]\}$. For each $i \in [1, \dots, N]$, let K_i denote the number of disjoint power intervals in $\mathcal{W}_i$. Observe that

$$K_i \leq K_{i-1} + 1 \tag{4.45}$$

Now, if $K_i = K_{i-1} + 1$, then $\mathcal{W}_i = \mathcal{W}_{i+1} \cup \{u_i\}$. Hence $\mathcal{V}_i = \{u_i\}$.

Otherwise $K_i \leq K_{i-1}$, u_i overlaps with at least one element of $\mathcal{W}_{i-1}$. Let $\mathcal{W}_i'$ denote the largest subset of $\mathcal{W}_{i-1}$ that overlaps with u_i, let w_i'' denote the combined user of $\mathcal{W}_i'$ and u_i. We know w_i'' is a member of $\mathcal{W}_i$. In addition, $\mathcal{V}_i = \mathsf{cmpl}(\mathcal{W}_i, \mathcal{W}_{i-1}) = \mathsf{cmpl}(w_i'', \mathcal{W}_i')$. Since $\mathcal{W}_{i-1}$ consists of disjoint power intervals, we conclude that $\mathsf{cmpl}(w_i'', \mathcal{W}_i')$ consists of no more than $K_{i-1} - K_i + 2$ power intervals. This is illustrated in Figure 4-15.

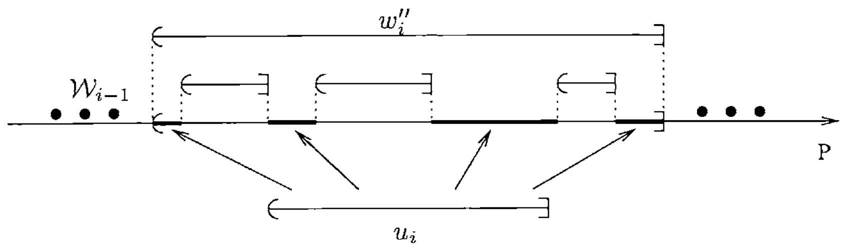


Figure 4-15: Constructing $\mathcal{V}_i$ when u_i overlaps with at least one member of $\mathcal{W}_{i-1}$ in Algorithm 4.2.3

Combining these two cases, the number of power intervals in $\mathcal{V}$ is upper bounded by

$$K_1 + \sum_{i=2}^{N} K_{i-1} - K_i + 2 = 2K_1 - K_N + 2(N-1) \tag{4.46}$$

Since $K_1 = 1$, and $\mathcal{U}$ has the overlapping property, i.e. $K_N = 1$, we have thus established that $\mathcal{V}$ contains no more than $2N - 1$ power intervals.

Incidentally, one may also reach this conclusion by noting that $\mathcal{V}$ is an adjacent set, and each computation of $\mathcal{V}_i$ creates no more than two additional end points (points marking the beginning and the ending of power intervals in $\mathcal{V}$) within $\mathsf{T}(\mathcal{U})$.

Four, we note that the member power intervals in $\mathcal{V}_i = \mathsf{cmpl}(\mathcal{W}_i, \mathcal{W}_{i-1})$ is bounded by the number of power intervals in $\mathcal{W}_{i-1}$ plus a constant. Since $K_1 = 1$, we observe that splitting-sets $\mathcal{V}_i$ with greater i tends to consist of more power intervals. In this sense, this construction algorithm may be used in multi-access communication systems in which transmitters are not equally adapted to the user-splitting technique, where transmitters less adapted are placed ahead in the ordering for the splitting algorithm.

Finally, recall that, from constructions of three overlapping power intervals discussed above, we were unable to fulfill the requirement that no user is split into more than 2 virtual users when an arbitrarily chosen user is not split. In fact, the user-splitting constructions resulting from this algorithm does not, in general, contain the construction established by Rimoldi and Urbanke in [12], where no user is split into more than 2 virtual users. To see this, consider the set of

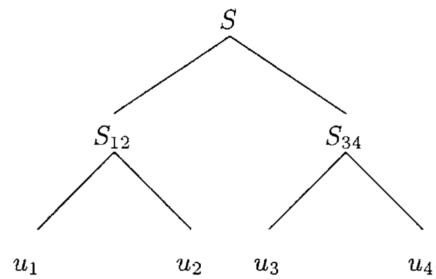


Figure 4-16: The only overlapping tree of $\mathcal{U} = \{u_i, i \in [1, \ldots, 4]\}$ satisfying conditions (4.47) through (4.50)

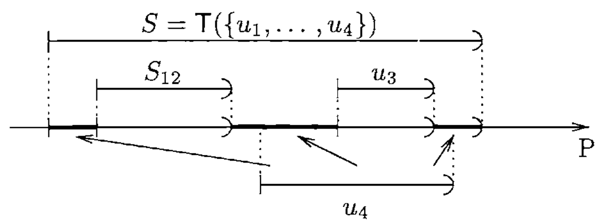


Figure 4-17: Illustrating the case with 4 overlapping power intervals, when at least one user is split into 3 virtual users

four overlapping power intervals $\mathcal{U} = \{u_i, i \in [1, \ldots, 4]\}$ where

$$u_1 \cap u_2 \neq \emptyset \tag{4.47}$$

$$u_3 \cap u_4 \neq \emptyset \tag{4.48}$$

$$\mathsf{T}(\{u_1, u_2\}) \cap \mathsf{ext}(\{u_3, u_4\}) = \emptyset \tag{4.49}$$

$$\mathsf{ext}(\{u_1, u_2\}) \cap \mathsf{T}(\{u_3, u_4\}) = \emptyset \tag{4.50}$$

Hence, they are the leaves of only the overlapping tree in Figure 4-16.

As illustrated in Figure 4-17, observe that no matter which order of users we choose to execute this algorithm, at least one user will be split into three virtual users.

One might try to modify this algorithm to include the constructions established in [12]. However, this turns out to be non-trivial, because the algorithm is, in a sense, "selfish". Specifically, during each iteration of the algorithm, the splitting-set for the user is constructed to accommodate only those previously constructed splitting-sets, without regard to the latter users.

To overcome this limitation, we will consider an algorithm that constructs the splitting-set for a user in an "unselfish" manner in each iteration. Specifically, instead of first allocating a splitting-set to the user, leaving the leftover to the remaining users, this algorithm first "reserves" a suitable set for the remaining users, and then assigns the leftover to be the splitting-set of the user. For such an algorithm to succeed, we need a simple check of the suitability of the "reserved" set to the remaining users. In other words, we need a simple answer to the suitability question without actually performing the dividing operations.

In the following sections, we will use the construction outcome of this algorithm to prove more general results concerning the suitability of sets of separated power intervals to given set of user specifications, i.e. whether the former may be divided into splitting-sets of the latter. Then, the results on suitability is combined with the conservation principle (to be defined) to produce all possible disjoint user-splitting constructions. In particular, we will exhibit the subset of that class that allows an arbitrarily chosen one user not to be split, and split no user into more than 2 virtual users.

4.3 Addressing The Suitability Question

In this section, we study the suitability of one set of separated power intervals for a set of user specifications.

For convenience, we make the following definition.

Definition 4.3.1. Let $\mathcal{W}$ denote a set of separated power intervals. Then $\mathsf{cmpl}(\mathsf{ext}(\mathcal{W}), \mathcal{W})$ consists of power intervals that are *holes* between the members of $\mathcal{W}$. $\mathsf{cmpl}(\mathsf{ext}(\mathcal{W}), \mathcal{W})$ is called the *hole set*, or the *holes* of $\mathcal{W}$, and denote it by $\mathsf{hole}(\mathcal{W})$.

Figure 4-18 illustrates this definition.

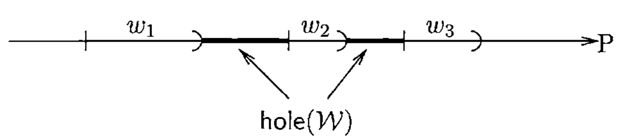


Figure 4-18: The *Holes* of a set of separated power intervals $\mathcal{W} = \{w_1, w_2, w_3\}$, $\mathsf{hole}(\mathcal{W})$

Now, the main theorem of this section.

Theorem 4.3.1. *Let $\mathcal{W}$ be a set of separated power intervals and $\mathcal{U}$ be a set of arbitrary power intervals such that $\mathsf{T}(\mathcal{W}) = \mathsf{T}(\mathcal{U})$. Then $\mathcal{W}$ is suitable to $\mathcal{U}$ if and only if $\mathcal{U} \cup \mathsf{hole}(\mathcal{W})$ has the overlapping property.*

Proof. Let $\mathcal{U} = \{u_i, i \in [1, \dots, N]\}$.

We prove the converse first. Suppose $\mathcal{W}$ is suitable to $\mathcal{U}$. By definition, there exists a division of $\mathcal{W}$, call it $\mathcal{W}'$, that is a splitting-set of $\mathcal{U}$. In particular, there exists a partition of $\mathcal{W}'$ where $\mathcal{W}' = \cup_{i=1}^{N} \mathcal{W}_i'$ such that the block $\mathcal{W}_i'$ resulting from the partition is the splitting-set of u_i for each i in $[1, \dots, N]$.

By the definition of complementing set, $\mathsf{hole}(\mathcal{W}) \cup \mathcal{W}$ is a division of $\mathsf{ext}(\mathcal{W})$. Hence, I: $\mathsf{hole}(\mathcal{W}) \cup \mathcal{W}$ has the overlapping property. This is by definition of the overlapping property.

II: $\mathsf{hole}(\mathcal{W}) \cup \mathcal{W}'$ also has the overlapping property. This is because $\mathcal{W}'$ is a division of $\mathcal{W}$, hence $\mathsf{hole}(\mathcal{W}) \cup \mathcal{W}'$ must also be a division of $\mathsf{ext}(\mathcal{W})$.

III: By Corollary 2.3.3, $\mathsf{hole}(\mathcal{W}) \cup \{\mathsf{T}(\mathcal{W}'_i), i \in [1, \ldots, N]\}$ has the overlapping property.

Finally, by the definition of splitting-set, $\mathsf{T}(\mathcal{W}'_i) = u_i$ for all $i \in [1, \ldots, N]$. The conclusion for this part follows.

For the forward part of the theorem, suppose $\mathsf{hole}(\mathcal{W}) \cup \mathcal{U}$ is an overlapping set. We need to show that $\mathcal{W}$ is suitable to $\mathcal{U}$.

Without the loss of generality, let $\mathsf{hole}(\mathcal{W}) = \{w_j, j \in [1, \ldots, M]\}$. Let μ be an ordering of the members of $\mathsf{hole}(\mathcal{W}) \cup \mathcal{U}$ such that every w_j is ordered ahead of every u_i for all $j \in [1, \ldots, M]$ and $i \in [1, \ldots, N]$. Now, apply Algorithm 4.2.3 to $\mathsf{hole}(\mathcal{W}) \cup \mathcal{U}$ with ordering μ. Observe that the splitting-sets for w_j are constructed before those of u_i for any $i \in [1, \ldots, N]$. Let the outcome of this algorithm be $\mathcal{V}$. Note that $\mathcal{V}$ is a division of $\mathsf{T}(\mathsf{hole}(\mathcal{W}) \cup \mathcal{U})$ and a splitting-set of $\mathsf{hole}(\mathcal{W}) \cup \mathcal{U}$.

Since $\mathsf{hole}(\mathcal{W})$ consists of separated power intervals, from the discussions on this construction algorithm in Section 4.2, recall that no w_i is split. Therefore $\mathsf{hole}(\mathcal{W})$ must be a subset of $\mathcal{V}$, i.e. $\mathsf{hole}(\mathcal{W}) \subset \mathcal{V}$. Let $\mathcal{W}' = \mathcal{V} \setminus \mathsf{hole}(\mathcal{W})$. Observe that $\mathcal{W}'$ is a splitting-set of $\mathcal{U}$.

Finally, since

$$\mathsf{T}(\mathsf{hole}(\mathcal{W}) \cup \mathcal{U}) = \mathsf{T}(\mathsf{hole}(\mathcal{W}) \cup \{\mathsf{T}(\mathcal{U})\}) = \mathsf{T}(\mathsf{hole}(\mathcal{W}) \cup \{\mathsf{T}(\mathcal{W})\}) = \mathsf{T}(\mathsf{hole}(\mathcal{W}) \cup \mathcal{W}) = \mathsf{ext}(\mathcal{W}) \tag{4.51}$$

$\mathcal{V}$ is a division of $\mathsf{ext}(\mathcal{W})$. Hence $\mathcal{W}'$ is a division of $\mathcal{W}$. This completes the proof. □

This theorem is the central result of this chapter. In the next section, we will study a class of disjoint user-splitting constructions based on this theorem. In particular, we will show that the resulting set of constructions subsumes all previously known such constructions. In fact, as we will see, the "only if" part of this theorem can be used to argue that the algorithm is capable of producing all possible disjoint splitting-set constructions.

Before leaving this section, we present a few simple consequences of this theorem for future use.

Recall that a set of N power intervals has the overlapping property if and only if the its irreducible equivalent consists of a single power interval, and the computation burden for constructing an irreducible equivalent is upper bounded by $O(N \ln N)$ (e.g. with Algorithm 2.3.8).

Suppose $\mathsf{hole}(\mathcal{W})$ consists of M power intervals, and $\mathcal{U}$ consists of N power intervals. Then the suitability check requires $O((N+M)\ln(N+M))$ computations.

Moreover, by the uniqueness of irreducible equivalents, we may compute $\mathsf{irr}(\mathsf{hole}(\mathcal{W}) \cup \mathcal{U})$ as $\mathsf{irr}(\mathsf{hole}(\mathcal{W}) \cup \mathsf{irr}(\mathcal{U}))$. We have the following corollary.

Corollary 4.3.2. *Let $\mathcal{W}$ denote a set of separated power intervals and $\mathcal{U}$ denote a set of arbitrary power intervals such that $\mathsf{T}(\mathcal{W}) = \mathsf{T}(\mathcal{U})$. Then $\mathcal{W}$ is suitable to $\mathcal{U}$ if and only if $\mathsf{irr}(\mathcal{U}) \cup \mathsf{hole}(\mathcal{W})$ has the overlapping property.*

The result of this corollary is useful when we are searching for a suitable set for $\mathcal{U}$, i.e. when we need to compute the irreducible equivalent of $\mathsf{hole}(\mathcal{W}) \cup \mathcal{U}$ for multiple $\mathcal{W}$, as it allows $\mathsf{irr}(\mathcal{U})$ to be pre-computed and thus reduces the computational burden for checking the suitability of $\mathcal{W}$.

Now, suppose $\mathcal{U}$ is an overlapping set. Then $\mathsf{irr}(\mathcal{U}) = \{\mathsf{T}(\mathcal{U})\}$. In this case, $\mathsf{T}(\mathcal{W}) = \mathsf{T}(\mathcal{U})$ implies that $\{\mathsf{T}(\mathcal{U})\} \cup \mathsf{hole}(\mathcal{W})$ has the overlapping property. Hence,

Corollary 4.3.3. *Let $\mathcal{W}$ denote a set of separated power intervals and $\mathcal{U}$ denote a set of overlapping power intervals such that $\mathsf{T}(\mathcal{W}) = \mathsf{T}(\mathcal{U})$. Then $\mathcal{W}$ is suitable to $\mathcal{U}$.*

From a more operational point of view, this corollary states that an overlapping set behaves identically as its combined user power interval, as in simplifying the AWGN multi-access achievability conditions. This completes the discussions in this section.

4.4 The Class of Conservation Algorithms

Given a set of user specifications $\mathcal{U}$, and a set of separated power intervals $\mathcal{W}$, Theorem 4.3.1 establishes a set of simple conditions for $\mathcal{W}$ to be suitable to $\mathcal{U}$, i.e. whether $\mathcal{W}$ can be divided into a splitting-set of $\mathcal{U}$, without actually dividing $\mathcal{W}$. In this section, we present a class of *conservation algorithms* which divides $\mathcal{W}$ into a splitting-set of $\mathcal{U}$ using results of this theorem. In fact, we will show that this class of algorithms is sufficiently general to produce all such possible division.

The class of conservation algorithms follows a resource allocation type of approach, where the *resource* is the suitable set of power intervals $\mathcal{W}$. Each iteration of a member algorithm in this class constructs a splitting-set for a single user u in $\mathcal{U}$ by allocating a portion of the available (remaining) resource guided by what we call the *conservation principle.* This principle is so named because instead of allocating a portion of the available resource for u directly (as in Algorithm 4.2.3), a suitable portion of the available resource is first "reserved" (or conserved) to the remaining users (users without splitting-sets assignments yet). Then, the leftover is assigned to be the splitting-set [5] of u, and is removed from the available resources for the next iteration. During each iteration, results of Theorem 4.3.1 is used to verify the suitability of the reserved portion to the remaining users.

Consider the following formalization of this class of algorithms.

Algorithm 4.4.1 (The Class of Conservation Algorithm).
Let a set of user power intervals $\mathcal{U}$ be arbitrarily ordered such that $\mathcal{U} = \{u_i, i \in [1, \ldots, N]\}$. Let $\mathcal{W}$ denote a set of separated power intervals suitable to $\mathcal{U}$. Initializing $\mathcal{W}_1 = \mathcal{W}$, we divide $\mathcal{W}$ into a splitting-set of $\mathcal{U}$ by performing the following constructions for each i from 1 through $N-1$:

Step 1: choosing $\mathcal{W}_{i+1}$, such that $\mathcal{W}_{i+1} \prec \mathcal{W}_i$, and $\mathcal{W}_{i+1}$ is suitable to $\{u_j, j \in [i+1, \ldots, N]\}$

Step 2: construct $\mathcal{V}_i = \mathsf{cmpl}(\mathcal{W}_i, \mathcal{W}_{i+1})$

Let $\mathcal{V}_N = \mathcal{W}_N$. Finally, let $\mathcal{V} = \cup_{i=1}^{N} \mathcal{V}_i$.

Note that this formalization does not specify how to choose $\mathcal{W}_{i+1}$ during each iteration. Indeed, this choice distinguishes individual algorithms in this class.

During each iteration, $\mathcal{W}_i$ is the set of available resources (intervals on the power axis), and $\mathcal{W}_{i+1}$ is the set of resource conceded to the remaining users. Regardless of how $\mathcal{W}_{i+1}$ is chosen, this algorithm completes successfully because each $\mathcal{W}_i$ was chosen in the previous iteration to be suitable to users $\{u_j, j \in [i, \ldots, N]\}$, which includes the current user u_i (note that for the first iteration, $\mathcal{W}_1$ is initialized to be suitable to $\mathcal{U}$), and $\mathcal{W}_{i+1}$ is chosen to be suitable to the remaining users $\{u_j, j \in [i+1, \ldots, N]\}$.

Observe that $\mathcal{W}_N$ is chosen to be a splitting-set of the last user. Hence no construction is necessary.

Finally, when the algorithm terminates, $\mathcal{V}$ is a splitting-set of $\mathcal{U}$ because $\mathsf{cmpl}(\mathcal{W}_i, \mathcal{W}_{i+1})$ in each iteration is a splitting-set of a user in $\mathcal{U}$. $\mathcal{V}$ is also a division of $\mathcal{W}$ because during each iteration, $\mathsf{cmpl}(\mathcal{W}_i, \mathcal{W}_{i+1}) \cup \mathcal{W}_{i+1}$ is a division of $\mathcal{W}_i$, and $\mathcal{W}_{i+1}$ is assigned to be the available resources for the next iteration.

To see the generality of the class of conservation algorithm, let $\mathcal{V}'$ be an arbitrary division of $\mathcal{W}$ that is also a splitting-set of $\mathcal{U}$. By definition of suitability, we may partition $\mathcal{V}'$ into blocks $\mathcal{V}'_i$ for each $i \in [1, \ldots, N]$ such that $\mathcal{V}'_i$ is a splitting-set of u_i. Observe that the union of the splitting-sets for all the remaining users during each iteration may be chosen as the

[5] The fact that the leftover is a splitting-set of u is guaranteed by Theorem 4.2.1.

conceded portion of available resources. The resulting $\mathcal{V}$ of such choices is $\mathcal{V}'$. Hence, every possible division of $\mathcal{W}$ that is a splitting-set of $\mathcal{U}$ can be produced by at least one member of the class of conservation algorithm (with appropriate choices of the conceded resources during each iteration).

From the generality of the class of conservation algorithm, observe that the same splitting-set outcome may be produced with arbitrary ordering of users in $\mathcal{U}$ by choosing $\mathcal{W}_{i+1}$ appropriately during each iteration. However, when the construction choice of the conceded resources during each iteration is fixed in advance, the ordering does matter to the outcome.

For example, observe that every possible construction outcome of Algorithm 4.2.3 can be produced by a distinct member of the class of conservation algorithms. Or, they can be produced by the same member with all possible ordering of members of $\mathcal{U}$, with the total available resource $\mathcal{W}$ being $\mathsf{irr}(\mathcal{U})$, and the available resource during each iteration $\mathcal{W}_i$ chosen to be the complementing set of the irreducible equivalent of $\{u_i, i \in [1, \dots, i]\}$ (all users with splitting-set assignments at the end of this iteration) in $\mathcal{W}$.

Finally, observe that this algorithm is possible because Theorem 4.3.1 offers a simple check of suitability prior to any further division operations.

We now present the construction of the sub-class of splitting-sets that satisfy the conditions set forth in [12]. Specifically, we are interested in splitting-sets that split no user into more than two virtual users, and leave at least one user un-split. Observe that such splitting-sets are desirable in multi-access communication systems in which transmitters are more or less equally adapted to the user-splitting technique. In addition, for a set of N user specifications, such a splitting-set consists of a total of no more than $2N-1$ virtual users.

Recall that, splitting-sets suitable to systems with transmitters of varying adaptability to user-splitting may be constructed using the simple splitting algorithm, the outcome of which consists of no more than $2N-1$ virtual users.

This sub-class of constructions may be described more simply using a variant of the class of conservation algorithm. Instead of following an ordering of users in $\mathcal{U}$ and constructing the splitting-set for a single user in each iteration, this variant recursively partitions the users into increasingly smaller blocks, each of which may consist of more than one user, and divides the increasingly smaller amount of available resources into suitable sets for the blocks.

Such recursive algorithms are particularly suited for sets of overlapping power intervals $\mathcal{U}$ that have the overlapping property, as an overlapping tree of $\mathcal{U}$ offers a natural scheme of recursively partitioning the users into increasingly smaller overlapping blocks. Moreover, Corollary 4.3.3 helps to simplify the check for suitability to minimal, as it states that the suitability of an overlapping set is identical to the suitability of a single power interval – the combined user of the overlapping set. We first consider $\mathcal{U}$ with the overlapping property.

Consider the following formalization.

Algorithm 4.4.2 (A Variant of the Conservation Algorithms).
Given an overlapping set of user power intervals $\mathcal{U} = \{u_i, i \in [1, \ldots, N]\}$, a set of separated power intervals $\mathcal{W}$ suitable to $\mathcal{U}$, and an overlapping tree of $\mathcal{U}$. For each node (leaf or intermediate) on the overlapping tree, let S denote the power interval at the node, and W_S denote the available resource for that node. Initializing all W_S except for the root node to be empty, initializing the available resource for the root node to be $\mathcal{W}$. Now, starting from the root node, the algorithm traverse through the entire tree by going to the immediate children of the current nodes next. At each intermediate node on the tree, the following are performed:

Step 1. *Let S denote the intermediate node of interest, let T_1 and T_2 denote the power intervals at the two immediate child nodes of S*

Step 2. *Choose $\mathcal{W}_{T_1}$ to be a splitting-set of T_1 such that $\mathcal{W}_{T_1} \prec \mathcal{W}_S$*

Step 3. *Construct $\mathcal{W}_{T_2} = \mathsf{cmpl}(\mathcal{W}_S, \mathcal{W}_{T_1})$*

Note that the operation of traversing through a tree may be accomplished using a recursion, i.e. by going to the child nodes next.

As in the formalization of the original conservation algorithms, note that this formalization does not specify how to choose the set $\mathcal{W}_{T_1}$ during each iteration. Indeed, this choice distinguishes individual algorithms specified in this variant.

Note that in this recursion, if $\mathcal{W}_S$ is suitable to S, then there exists one appropriate choice of $\mathcal{W}_{T_1}$ by Corollary 4.3.3. By Theorem 4.2.1, $\mathcal{W}_{T_2} = \mathsf{cmpl}(\mathcal{W}_S, \mathcal{W}_{T_1})$ is suitable to T_2. Since the recursion is started at the root node with $\mathcal{W}_S = \mathcal{W}$, a suitable set to the root node power interval $\mathsf{T}(\mathcal{U})$, we conclude that the available resources at every node on the tree is a splitting-set of the power interval at the node. In particular, $\mathcal{W}_{u_i}$ constructed is a splitting-set of u_i for each $i \in [1, \ldots, N]$. Hence $\cup_{i=1}^{N} \mathcal{W}_{u_i}$ is a splitting-set of $\mathcal{U}$. Let $\mathcal{V} = \cup_{i=1}^{N} \mathcal{W}_{u_i}$.

$\mathcal{V}$ is a division of $\mathcal{W}$ follows from the construction $\mathcal{W}_{T_2} = \mathsf{cmpl}(\mathcal{W}_S, \mathcal{W}_{T_1})$, which implies that $\mathcal{W}_{T_1} \cup \mathcal{W}_{T_2}$ is a division of $\mathcal{W}_S$ (by definition of complementing set) for each intermediate node S. Since $\mathcal{W}_S$ at the root node is $\mathcal{W}$, we may carry this implication from the root node down to reach the desired conclusion.

In addition, observe that a binary tree of N leaves consists of $N-1$ intermediate nodes. Hence no more than $N-1$ full iterations of the recursion is necessary for a set of N overlapping users. Hence the computational burden of this variant is similar to the conservation algorithm.

Finally, if multiple overlapping trees exist for the same $\mathcal{U}$, observe that, when the construction of $\mathcal{W}''$ is chosen to be sufficiently general, any one tree may be used to produce all possible constructions. However, for specific constructions of $\mathcal{W}'$, the choice of the overlapping tree may affect the resulting construction.

To avoid any confusion, let me emphasize that we have shown that the class of conservation algorithm are sufficiently general to produce all possible constructions with which we are concerned here. So this variant does not produce any new construction, and we present it only because it better displays the insights leading to a splitting-set in which no user is split into more than 2 virtual users, and at least one user is not split.

We approach such user-splitting construction by choosing $\mathcal{W}_{T_1}$, from a $\mathcal{W}_S$ consisting of no more than 2 disjoint power intervals, such that neither $\mathcal{W}_{T_1}$ nor $\mathcal{W}_{T_2}$ consists of more than 2 virtual users. When $\mathcal{U}$ has the overlapping property, we may choose $\mathcal{W} = \mathsf{T}(\mathcal{U})$. We may then combine the variant algorithm with the above choice of $\mathcal{W}_{T_1}$ to arrive at a construction such that no $\mathcal{W}_S$ on the overlapping tree consists of more than 2 disjoint power intervals, thereby accomplishes our objective.

In the following, we present such a construction choice of $\mathcal{W}_{T_1}$ in three cases. For simplicity of description, note that both T_1 and T_2 must be contained in $\mathsf{ext}(\mathcal{W}_S)$[6].

Case a: If $\mathcal{W}_S$ consists of a single power interval, this power interval must be S as $\mathcal{W}_S$ is a splitting-set of S. This case is identical to the splitting of two overlapping users introduced in Figure 4-7 discussed in the introduction of this chapter.

We detail the constructions in this case as follows: choose $\mathcal{W}_{T_1} = \{T_1\}$. Then $\mathcal{W}_{T_2} = \mathsf{cmpl}(\mathcal{W}_S, \mathcal{W}_{T_1}) = \mathsf{cmpl}(S, T_1)$. Figure 4-19 illustrates the construction in this case. Observe that $\mathcal{W}_{T_1}$ is a singleton, and $\mathcal{W}_{T_2} = \mathsf{cmpl}(S, T_1)$ may not consists of more than two separated power intervals.

Case b: Suppose $\mathcal{W}_S$ consists of two separated power intervals, and T_1 is not contained in either member of $\mathcal{W}_S$. Then T_1 must intersect with $\mathsf{hole}(\mathcal{W}_S)$. The construction for this case is identical to the three-overlapping-user case illustrated in Figure 4-11.

We detail the constructions in this case as follows: first note $\mathsf{hole}(\mathcal{W}_S) \cup \{T_1\}$ forms an overlapping set, hence $T_1 \subseteq \mathsf{T}(\mathsf{hole}(\mathcal{W}_S) \cup \{T_1\})$. Therefore T_2 must intersect with $\mathsf{T}(\mathsf{hole}(\mathcal{W}_S) \cup \{T_1\})$. By Theorem 2.3.4, $\{T_1, T_2\} \cup \mathsf{hole}(\mathcal{W}_S)$ has the overlapping

[6] To see this, first observe that by definition of splitting-set, $\mathsf{T}(\mathcal{W}_S) = S$. Then, since $\mathcal{W}_S$ consists of separated power intervals, by Corollary 2.2.4, we have $S \subseteq \mathsf{ext}(\mathcal{W}_S)$. Finally, since T_1 and T_2 overlap, and their combined user is S, by definition of the overlapping property, both T_1 and T_2 must be contained in S. The desired conclusion follows.

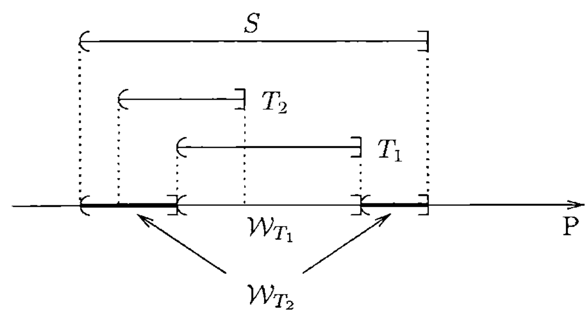


Figure 4-19: Case a

property. By the definition of the overlapping property,

$$\mathsf{ext}(\mathsf{hole}(\mathcal{W}_S)) \subseteq \mathsf{T}(\mathsf{hole}(\mathcal{W}_S) \cup \{T_1\}), \mathsf{hole}(\mathcal{W}_S) \subseteq \mathsf{T}(\{T_1, T_2\} \cup \mathsf{hole}(\mathcal{W}_S)) \quad (4.52)$$

Take $\mathcal{W}_{T_1} = \mathsf{cmpl}(\mathsf{T}(\mathsf{hole}(\mathcal{W}_S) \cup \{T_1\}), \mathsf{hole}(\mathcal{W}_S))$. By Theorem 4.2.1, $\mathcal{W}_{T_1}$ is a splitting-set of T_1. Hence

$$\mathcal{W}_{T_2} = \mathsf{cmpl}(\mathsf{T}(\{T_1, T_2\} \cup \mathsf{hole}(\mathcal{W}_S)), \mathsf{T}(\mathsf{hole}(\mathcal{W}_S) \cup \{T_1\})) \quad (4.53)$$

Observe that $\mathsf{hole}(\mathcal{W}_s)$ is contained in both parameter of the complementing operations. Figure 4-20 illustrates this construction. Observe that the splitting-set assigned to neither T_1 nor T_2 consists of more than two power intervals.

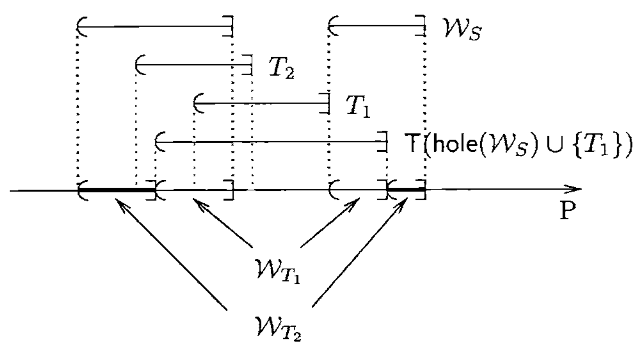


Figure 4-20: Case b

Case c: Otherwise, $\mathcal{W}_S$ consists of two separated power interval, and T_1 is contained in a member of $\mathcal{W}_S$[7]. Without the loss of generality, let $T' \in \mathcal{W}_S$ be such that $T_1 \subseteq T'$. Observe that $\mathsf{cmpl}(T', T_1) \cup \{T_1\}$ is an adjacent set, and hence has the overlapping property. By Theorem 2.3.2, $\{T_1, \mathsf{T}(\mathsf{cmpl}(T', T_1))\}$ has the overlapping property. In particular, the combined user of $\{T_1, \mathsf{T}(\mathsf{cmpl}(T', T_1))\}$ is T'. Choose $\mathcal{W}_{T_1} = \mathsf{cmpl}(T', \mathsf{T}(\mathsf{cmpl}(T', T_1)))$.

[7]Note that this is the only new constructions introduced here.

By Theorem 4.2.1, $\mathcal{W}_{T_1}$ is a splitting-set of T_1. Observe that $\mathcal{W}_{T_1}$ consists of two power intervals. Additionally, observe $\mathcal{W}_{T_2} = \mathsf{cmpl}(\mathcal{W}_S, \mathcal{W}_{T_1})$ consists of $\mathsf{T}(\mathsf{cmpl}(T', T_1))$ and the other member of $\mathcal{W}_S$. Figure 4-21 illustrates this construction.

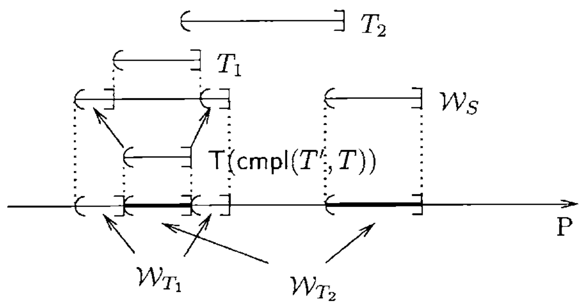


Figure 4-21: Case c

Observe that the total available resources $\mathcal{W}$ is given to be $\mathsf{T}(\mathcal{U})$, which is a singleton set. Then Case a is used for the first iteration, and for the next iteration with T_1 and its splitting-set, and so on and so forth. Consequently, the splitting-set of at least one user is a singleton (i.e. unsplit). Hence, the resulting $\mathcal{V}$ may not consist of more than $2N - 1$ virtual users.

Note that these construction choices lead to all the user-splitting constructions presented in [12].

However, this construction of $\mathcal{W}_{T_1}$ is by no mean the only one which, when used in the variant algorithm, produces a splitting-set $\mathcal{V}$ such that no user in $\mathcal{U}$ is split into more than 2 virtual users. For example, under certain conditions, the construction illustrated in Figure 4-22 may be used for both Case b or Case c. Additionally, as discussed in the introduction of this chapter, splitting-sets need not be divisions of the irreducible equivalent of the set of users (particularly as we consider general sets of user specifications below). Finally, it is not necessary to require each iteration in the variant algorithm to produce splitting-sets with no more than 2 virtual users to achieve the desired final outcome. As all constructions here may be performed graphically on the power axis, readers are encouraged to explore further such possibilities.

Finally, for a general non-overlapping set of N user specifications $\mathcal{U}$, we may again partition $\mathcal{U}$ into maximally overlapping blocks using the irreducible equivalent construction. Then, we may apply the variant algorithm along with the construction choice of $\mathcal{W}_{T_1}$ above to each constructing sets of the irreducible equivalent. And finally, since the combined users of the maximally overlapping subsets are separated, we may combine the outcome for the overlapping subsets to arrive at a splitting-set of $\mathcal{U}$ in which no member of $\mathcal{U}$ is split into more than 2 virtual users, and at least M users are not split, where M is the number of maximally overlapping blocks in the irreducible equivalent. Note again, that the total number of power intervals in the resulting splitting-set is strictly fewer than the construction established by Rimoldi and Urbanke

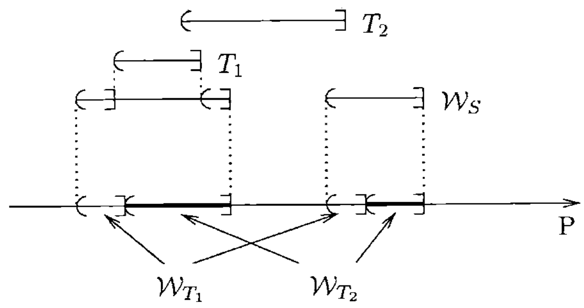


Figure 4-22: Another possible construction of $\mathcal{W}_{T_1}$

in [12] for rate tuples in the interior of the AWGN multi-access achievable rate region.

With this, we complete the discussion on successive decoding with user-splitting.

Chapter 5

The Time-Power Diagram and Time-Sharing in AWGN Multi-Access Channels

Time-sharing was first applied to AWGN multi-access channels for achieving rate tuples on the dominant face of the achievable rate region (c.f. [11], [8]). We illustrate this usage with the two-user case. Let n_o denote the channel noise variance, and for $i \in 1, 2$, let p_i and r_i respectively be the power constraint and rate specifications of user i.

Recall that, as illustrated in Figure 5-1, the *dominant face* of the achievable rate region of this channel is the line segment between the two corner rate tuples (corner points) R_1 and R_2, where

$$\mathsf{R}_1 = \left(\frac{1}{2}\ln\left(\frac{n_o + p_1}{n_o}\right), \frac{1}{2}\ln\left(\frac{n_o + p_1 + p_2}{n_o + p_1}\right)\right) \tag{5.1}$$

$$\mathsf{R}_2 = \left(\frac{1}{2}\ln\left(\frac{n_o + p_1 + p_2}{n_o + p_2}\right), \frac{1}{2}\ln\left(\frac{n_o + p_2}{n_o}\right)\right) \tag{5.2}$$

Observe that the line segment between two points can be written as the convex combination of the two points. In other words, for each rate tuple $\mathsf{R} = (r_1, r_2)$ on this dominant face, there exists a $\lambda \in [0, 1]$ such that

$$\mathsf{R} = \lambda \mathsf{R}_1 + (1 - \lambda)\mathsf{R}_2 \tag{5.3}$$

Hence, one may achieve R by synchronizing the two transmitters to transmit using rate tuple R_1 for a portion λ of time, and to transmit using rate tuple R_2 for the remaining $1 - \lambda$ portion of time.

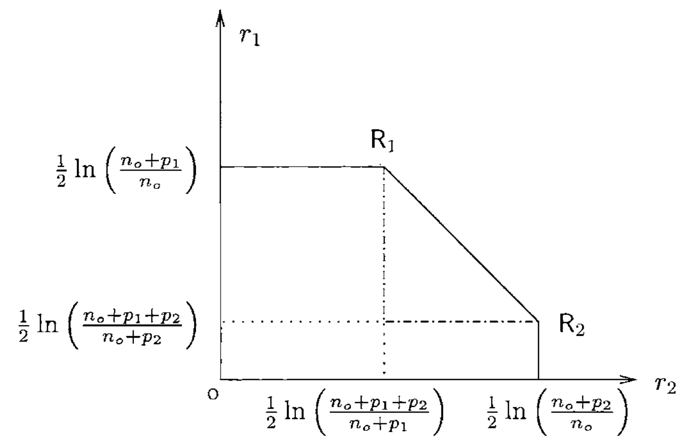


Figure 5-1: Time-sharing in a 2-user AWGN multi-access channel

This scheme can be easily generalized to the N-user case. Recall that the dominant face of the N-user AWGN multi-access achievable rate region is the convex hull of its corner points. By definition of convex hull, every rate tuple on the dominant face of an AWGN multi-access achievable rate region can be written as a convex combination of the corner points. Recall that there are $N!$ corner points, each corresponding to a distinct successive decoding order. Let R_μ denote the rate tuple at the corner point corresponding to decoding order μ. Let R be a rate tuple on the dominant face, we therefore have $\lambda_\mu \in [0,1]$ for each $\mu \in \mathbf{\Pi}(N)$ such that

$$\mathsf{R} = \sum_{\mu \in \mathbf{\Pi}(N)} \lambda_\mu \mathsf{R}_\mu \tag{5.4}$$

where $\sum_{\mu \in \mathbf{\Pi}(N)} \lambda_\mu = 1$. Hence, one may achieve R by synchronizing all N transmitters in the system to transmit using rate tuple R_μ for a portion λ_μ of time for each $\mu \in \mathbf{\Pi}(N)$[1].

Such a technique has been termed *time-sharing*. Note that the difference between time-sharing and time-division multiple access (TDMA) is that only one user is allowed to transmit during each channel use in TDMA.

For simplicity, we will refer to λ_μ as the *utility portion* of the rate tuple R_μ for achieving R.

One may implement this time-sharing scheme in discrete time AWGN multi-access channel is as follows:

1. Section all channel uses into frames each consisting of K channel uses

2. Choose an ordering of all elements of $\mathbf{\Pi}(N)$, call it f, such that $f(i)$ denote a unique $\mu \in \mathbf{\Pi}(N)$ for each $i \in [1, \dots, N!]$

3. Within each frame of K channel uses, synchronize all N transmitters in the system to

[1] Of course, we may skip R_μ for $\lambda_\mu = 0$.

transmit at rate tuple $\mathsf{R}_{f(i)}$ between channel uses l_{i-1} and $l_i - 1$, where

$$l_0 = 0 \tag{5.5}$$

$$l_i = \left\lfloor K \times \sum_{j=1}^{i} \lambda_{f(i)} \right\rfloor, i \in [1, \ldots, N] \tag{5.6}$$

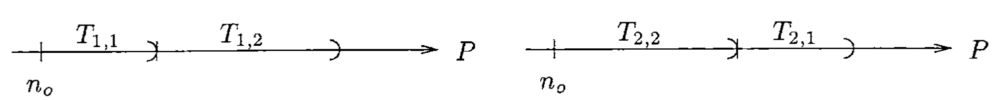
and $\lfloor x \rfloor$ denote the greatest integer less than x.

Thus the transmission pattern is repeated in every block of K channel uses. Observe that R is achieved asymptotically by choosing K sufficiently large.

Note that in this implementation, λ_μ designates the portion of channel uses for which R_μ is used. For this reason, we will also refer to λ_μ as the *time portion* for R_μ.

We observe that there are multiple implementations of time-sharing. We chose this one because it leads to a simple correspondence to the time-power diagram framework on which we will develop our understandings. We note that modifications can be made in our discussion (on the conceptual correspondence) to relate the time-power diagram framework to other implementations of time-sharing.

One advantage of this time-sharing scheme is that the receiver may successively decode the transmissions. To see this, recall that in Chapter 2, we showed that each corner point on the dominant face corresponds to a set of adjacent power intervals among which the least of the lower boundary is the channel noise variance. To illustrate this for the two-user case, let $T_{j,i}$ denote the power interval of user i at R_j, where $i, j \in \{1, 2\}$. The power diagrams for the two corner points on this two-user AWGN multi-access achievable rate region are illustrated in Figure 5-2.



(a) Power diagram for R_1 (b) Power diagram for R_2

Figure 5-2: Power diagrams for corner points of a two-user AWGN multi-access channel

In Chapter 3, we showed that adjacent power intervals are successively decodable following the descending order of the power intervals. Hence, the decoder, knowing the rate tuple for each channel use, may first group the transmissions (discrete-time channel uses) according to the rate tuple. Then each group of transmissions is successively decoded. Finally, the decoding results from all the groups are appropriately recombined to recover the users' messages.

In this two-user example, note that the first group of transmissions (which corresponds to rate tuple R_1) may be successively decoded with decoding order $\{2,1\}$. In particular, $T_{1,2}$ is decoded first, then the transmission of user 2 in this group is stripped, and $T_{1,1}$ is decoded. Similarly, the second group of transmissions (which corresponds to rate tuple R_2) may be successively decoded with decoding order $\{1,2\}$. Finally, the two decoded messages of each user (one in both groups) are appropriately combined to recover their original messages. We are thus able to successively decode a rate tuple on the dominant face of the 2-user AWGN multi-access achievable rate region. From Chapter 3, recall that such a set of user specifications is not successively decodable with either simple successive decoding or power-reduced successive decoding.

This scheme can be simply extended to the general N-user case with (5.4). Moreover, since every rate tuple in the interior of the AWGN multi-access achievable rate region is componentwise dominated by at least one rate tuple on the dominant face, we arrive at the simple conclusion that the maximal noise threshold in an AWGN multi-access channel in which a given set of user specifications is successively decodable with time-sharing is the maximal noise threshold for the set of users to be achievable.

We note that in theory, the complexity of successive decoding is determined by the number of decoding steps required[2] Our focus in studying successive decoding with time-sharing is on reducing the number of required successive decoding steps.

In the two-user example above, note that a total of 4 successive decoding steps are necessary.

Alternatively, once user 2's transmission in the group corresponding to R_1 (which corresponds to power interval $T_{1,2}$) is decoded, user 1's transmission in both groups ($T_{1,1}$ and $T_{2,1}$) may be decoded together in a single successive decoding step with maximum *a posteriori* decoding. This may be accomplished by appropriately accounting for the combined channel noise and the interference from the remaining transmissions of user 2 in the other group of transmissions corresponding to R_2 ($T_{2,2}$).

In this decoding scheme, the receiver regards user 1's transmission in the group corresponding to R_2 ($T_{2,1}$) as if it is transmitted through a channel with interference variance $n_o + p_2$, which is the sum of the channel noise and user 2's transmission. From this point-of-view, it is as if user 1 transmits in one of two parallel and independent channels for each channel use, and the knowledge of which channel is used during each transmission is known to both the

[2]We also note, that such may not be the case in real systems when neither the block length of transmission codes used is arbitrarily large, nor the probability of decoding error is arbitrarily small. We additionally note that it is entirely possible that larger number of decoding steps might make the decoding in each successive decoding step much easier in real systems.

transmitter and the receiver.

Finally, after the first user's message in both groups is decoded and stripped, the second user's transmission in the group corresponding to R_2 ($T_{2,2}$) can be decoded, and the two decoding result of user 2 is then combined to recover the first user's message. Such decoding schemes were first recognized and proposed by Yeh and Gallager [18].

Observe that this scheme, at least in this two-user case, has the advantage of requiring 3 successive decoding steps, which is the same as that required by the *user-splitting* technique. Compared to the previous decoding scheme, the complexity of the decoder for the first user is increased slightly, but the error exponent from coding the two transmissions together may be improved.

In [18], Yeh and Gallager studied both decoding schemes for general N-user AWGN multi-access channels. They noted that the dominant face of the N-user AWGN multi-access achievable rate region is a convex region in an N dimensional space, and hence any rate tuple on this dominant face may be written as the convex combination of no more than N corner points. In other words, they showed that using (5.4), no more than N λ_μ's are required to be non-zero[3]. In addition, they showed that using the second decoding scheme, the total number of decoding steps required grows as $N + \frac{1}{2}N\log(N)$.

In this chapter, we approach time-sharing by extending the power diagram framework into the *time-power diagram* framework. This new framework is the result of grafting a new axis to the power diagram. We are thus able to visually represent all the sets of user specifications used in time-sharing, along with their respective time portions, together in a single two-dimensional space. In the following, we will use the time-power diagram to study various time-sharing constructions that achieve a given set of user specifications. In particular, we will establish that no more than $2N-1$ successive decoding steps are necessary to decode every rate tuple in the N-user AWGN multi-access achievable rate region with time-sharing[4]. This result improves the one in [18], and establishes the *time-sharing* technique as an equivalent alternative to *user-splitting*.

The rest of this chapter is organized into 2 sections. We first introduce the time-power diagram in Section 5.1, and discuss our approach to designing time-sharing constructions that requires no more than $2N-1$ successive decoding steps. Then, in Section 5.2, we discuss the

[3]When formulated in this manner, this result may also be seen as a direct consequece of properties of convex hull of finite number of points in N dimensional spaces

[4]As mentioned in the introduction chapter that, as we approach the completion of this thesis, we become aware of a work recently submitted by Rimoldi for publication that reduces the number of successive decoding steps to $O(N)$ with time-sharing.

strategy to accomplish our objective using the time-power diagram.

5.1 Introducing the Time-Power Diagram

In the above, we have seen that time-sharing is a technique that allows different user specifications to be used for different channel uses. In other words, the power diagram may vary from channel use to channel use. The time-power diagram framework is a way to capture such variations and visually display them together. Following the implementation model of time-sharing in the above, we only need to specify the power diagrams for each channel use in the K-block. From this point of view, the time-power diagram can be seen as a way of stacking the K power diagrams up vertically along a new axis, which we call the *time axis*. More specifically, we grow each of the power diagrams into a horizontal strip of width $1/K$, which is the time portion of the power diagram for a single channel use in the K-block. Hence the power interval of each user on each power diagram is grown into a rectangle of width $1/K$. We then align the power axes, and place them side by side.

We use the two-user case above to illustrate. Following the time-sharing implementation in the above, in each K-block, both users transmit following specifications given in the power diagram in Figure 5-2(a) for channel uses 0 through $\lfloor K\lambda \rfloor$, and following that in Figure 5-2(a) for channel uses $\lfloor K\lambda \rfloor + 1$ through $K - 1$. This is illustrated in Figure 5-3.

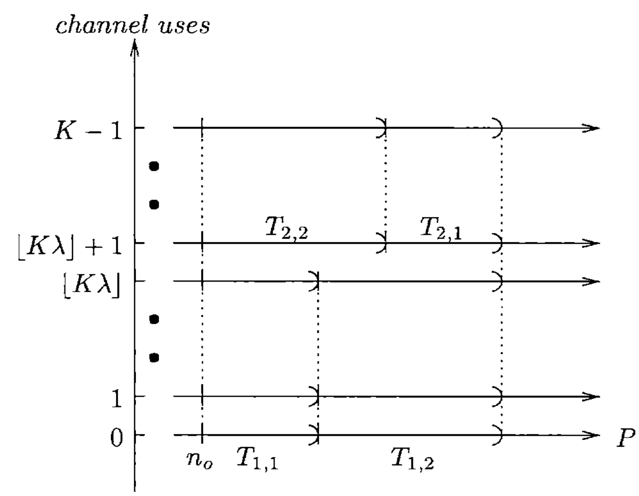


Figure 5-3: Illustrating the construction of the time-power diagram with the 2-user case: stacking the K power diagrams

Taking K sufficiently large, scaling the channel use axis so that the K-block is normalized to have a unit length, the outcome approaches the time-power diagram depicted in Figure 5-4.

Definition 5.1.1. Let a user be specified to transmit at power and rate specifications given by

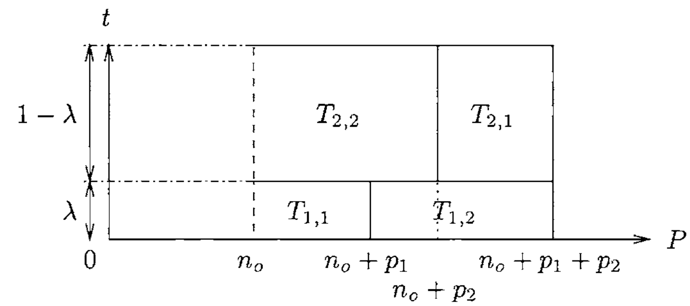


Figure 5-4: Representing time-sharing in a 2-user AWGN multi-access channel using the time-power diagram

power interval T for channel uses number $\lfloor \alpha K \rfloor + 1$ and $\lfloor (\alpha+\lambda)K \rfloor$ in each K-block for any $K > 0$, where $0 \leq \alpha, \alpha+\lambda \leq 1$. The *time-power bi-interval* representation of this specification is $T \times (\alpha, \alpha+\lambda]$, and $T \times (\alpha, \alpha+\lambda]$ is said to be a *time-power interval* of this user. T and $\mathcal{A}$ are respectivelly called the *power interval* and *time interval* of this time-power bi-interval.

Observe that the length of time interval is the time portion. Observe that $T_{1,1} \times (0, \lambda]$ is a time-power bi-interval of user 1. So is $T_{1,1} \times \mathcal{A}$ for any $\mathcal{A} \subseteq (0, \lambda]$.

Definition 5.1.2. Given a user's time-power bi-interval $\widehat{T} = T \times \mathcal{A}$[5],

- let $\widehat{\mathsf{I}}_p(\widehat{T})$ denote its power interval, T
- let $\widehat{\mathsf{I}}_t(\widehat{T})$ denote its time interval, $\mathcal{A}$

Definition 5.1.3. The *rate contribution* of a given time-power bi-interval Tt is $\lambda \mathsf{r}\left(\widehat{\mathsf{I}}_p(\widehat{T})\right)$, where λ is the length of $\widehat{\mathsf{I}}_t(\widehat{T})$. We denote this quantity by $\widehat{\mathsf{r}}(\widehat{T})$.

Observe that if a user transmits at a set of constant specifications for all K channel uses, then the time interval of its time-power bi-interval for this specification is $(0, 1]$. Note that this corresponds to the case of no time-sharing. In the following, when the discussion makes it clear that a time-power interval is concerned, we may abbreviate a time-power bi-interval $T \times (0, 1]$ as T for simplicity. Conversely, we will say that the time interval of a power interval is $(0, 1]$, i.e.

$$\widehat{\mathsf{I}}_t(T) = (0, 1] \tag{5.7}$$

Observe that, by definition of time-power bi-intervals, if a set of users are specified to transmit at power and rate specifications given by the set of power intervals $\mathcal{U}$ for channel uses

[5]For easy of distinction, we will use $tNote$ to designate variables and functions in the time-power diagram framework.

number $\lfloor \alpha K \rfloor + 1$ and $\lfloor (\alpha + \lambda)K \rfloor$ in each K-block for any $K > 0$, then the time-power bi-interval representation of the specifications of every user in $\mathcal{U}$ must have time interval $(\alpha, \alpha + \lambda]$. For simplicity of notation in such cases, we define the following.

Definition 5.1.4. Let $\mathcal{U}$ be a set of power intervals, and $\mathcal{A}$ be a time interval. Then, we use $\mathcal{U} \times \mathcal{A}$ to denote the set of time-power bi-intervals

$$\{u \times \mathcal{A}, \forall u \in \mathcal{U}\} \tag{5.8}$$

Definition 5.1.5. In each K-block for any $K > 0$, let the specifications of a set of users for channel uses number $\lfloor \alpha K \rfloor + 1$ through $\lfloor (\alpha + \lambda)K \rfloor$ be specified by the set of power intervals $\mathcal{U}$. Then the set of time-power bi-intervals $\mathcal{U} \times (\alpha, \alpha + \lambda]$ is said to be a *coherent segment.* The time interval $(\alpha, \alpha + \lambda]$ is said to be the *time specification* of the coherent segment, and $\mathcal{U}$ is said to be the *user specifications* of the coherent segment.

In particular, the time interval of every time-power bi-interval in a coherent segment with time specification $\mathcal{A}$ is identically $\mathcal{A}$. Note that, graphically, this coherent segment is equivalent to a horizontal strip of the time-power diagram (corresponding to a time interval) which does not contain any boundary indicating changing user specifications in time.

Note that the user specifications in a coherent segment effectively describes the user specifications in each one of the group of transmissions considered for successive decoding in the introduction section.

Observe that there are two coherent segments in the time-power bi-interval representation of the two-user case illustrated above: one specified by time interval $(0, \lambda]$, and the other by $(\lambda, 1]$. Note that the entire time-power bi-interval representation in this case can be seen as the union of the two coherent segments, $\{T_{1,1}, T_{1,2}\} \times (0, \lambda]$, and $\{T_{2,1}, T_{2,2}\} \times (\lambda, 1]$. These definitions make it possible to regard this case as *time-sharing between the user specifications of the* 2 *coherent segments.* This perspective allows us to approach time-sharing by

> first constructing a set of time-power bi-intervals for a given set of user specifications; and then collecting members of the set of time-power bi-intervals into coherent segments.

In this approach, note that the number of sets of user specifications used in time-sharing is identical to the number of coherent segments.

Note that the choice of coherent segments may not be unique. For example, in the two-user case above, $\{T_{1,1}, T_{1,2}\} \times \mathcal{A}$ for any $\mathcal{A} \subseteq (0, \lambda]$ is also a coherent segment. Furthermore, note that the above discussions on time-sharing allow for time-sharing between two identical sets of user power and rate specifications. Therefore, one might consider representing this two-user case by a set of time-power bi-intervals consisting of 3 coherent segments, $\{T_{1,1}, T_{1,2}\} \times (0, \alpha]$, $\{T_{2,1}, T_{2,2}\} \times (\alpha, \alpha + 1 - \lambda]$, and $\{T_{1,1}, T_{1,2}\} \times (\alpha + 1 - \lambda, 1]$, with α arbitrarily chosen in $(0, \lambda]$. Observe that this new representation specifies that three sets of user specifications are used instead of 2 as in Figure 5-4. Such a specification is no different from reserving a subset of required channel uses for the specification governed by some rate tuple (in this case, R_1) to be transmitted at a later time in the K-block.

For simplicity, in the rest of this document, we will always re-order the channel uses in the K-block, i.e. swapping the relative location of the time interval of coherent segments, such that those coherent segments with identical set of user specifications are represented by adjacent time intervals, and combined into a single coherent segment. Hence, when referring to coherent segments by their user specifications, distinct coherent segments correspond to distinct sets of user specifications.

From the above, recall that each user is restricted to transmit with the same amount of power using a single transmission codebook during every channel use in time-sharing. In other words, every user can only have a single time-power bi-interval in each coherent segment, and the power specifications of a user's time-power interval in every coherent segment are identical.

Definition 5.1.6. Let $\widehat{\mathcal{W}}$ be a set of time-power bi-intervals such that the power interval of each member of $\widehat{\mathcal{W}}$ has identical power specifications, and the time intervals of all members of $\widehat{\mathcal{W}}$ form an adjacent set with extent equal to $(0, 1]$. We shall say that such a set has the *simple* property.

The *power specification* of $\widehat{\mathcal{W}}$ is the power specification of its member time-power bi-intervals, denote it by $\overline{\mathsf{p}}\left(\widehat{\mathcal{W}}\right)$.

The *rate specification* of $\widehat{\mathcal{W}}$ is the combined rate contribution of its member time-power bi-intervals, denote it by $\overline{\mathsf{r}}\left(\widehat{\mathcal{W}}\right)$. We have

$$\overline{\mathsf{r}}\left(\widehat{\mathcal{W}}\right) = \sum_{\widehat{w} \in \widehat{\mathcal{W}}} \widehat{\mathsf{r}}(\widehat{w})$$

Definition 5.1.7. Let a user's power and rate specifications be given by power interval T. Let $\widehat{\mathcal{U}}$ denote a set of time-power bi-intervals with the simple property. Suppose

$$\overline{\mathsf{p}}\left(\widehat{\mathcal{U}}\right) = \mathsf{p}(T) \tag{5.9}$$

$$\overline{\mathsf{r}}(\widehat{\mathcal{U}}) = \mathsf{r}(T) \tag{5.10}$$

Then $\widehat{\mathfrak{U}}$ is said to be an *achieving set* of this user.

In the two-user case above, we let

$$\widehat{T}_{1,1} = T_{1,1} \times (0, \lambda] \tag{5.11}$$

$$\widehat{T}_{2,1} = T_{2,1} \times (\lambda, 1] \tag{5.12}$$

$$\widehat{T}_{1,2} = T_{1,2} \times (0, \lambda] \tag{5.13}$$

$$\widehat{T}_{2,2} = T_{2,2} \times (\lambda, 1] \tag{5.14}$$

Note that $\left\{\widehat{T}_{1,1}, \widehat{T}_{2,1}\right\}$ is user 1's achieving set, and $\left\{\widehat{T}_{1,2}, \widehat{T}_{2,2}\right\}$ is user 2's achieving set.

We are now able to formalize the following.

Definition 5.1.8. A set of time-power bi-intervals $\widehat{\mathfrak{U}}$ is said to be a *time-sharing scheme* of a set of N user specifications $\mathcal{U}$ if there exists a partition of $\widehat{\mathfrak{U}}$ and a one-to-one correspondence between the blocks of the partition and members of $\mathcal{U}$ such that each block is the achieving set of the corresponding user.

For example, the set of four time-power bi-intervals in the two-user case above

$$\left\{\widehat{T}_{1,1}, \widehat{T}_{1,2}, \widehat{T}_{2,1}, \widehat{T}_{2,2}\right\}$$

forms a time-sharing scheme of the set of two-user specifications.

In the introduction to this chapter, we have seen that no more than N corner points are required to achieve any rate tuple on the dominant face with time-sharing because of the convexity of the dominant face of the N-user AWGN multi-access achievable rate region. Using the terminology developed here, this observation is equivalent to stating that, given a set of N user specifications that has the overlapping property, there exists time-sharing schemes with no more than N coherent segments.

In successive decoding with time-sharing, parts of a user's transmission (at a number of channel uses which are specified in a number of coherent segments) are decoded in a single successive decoding step. Since the amount of interference (from the remaining users' transmissions) in different coherent segments may differ, completing this step requires the user's transmission in each of the coherent segment to be decodable, in the presence of their respective interference. This leads to the conclusion that the set of user specifications in each coherent segment must be successively decodable. For convenience, we will say that such a coherent segment is *successively decodable*.

Recall, from Chapter 3, that the procedure of successively decoding and stripping user's transmissions can be visualized, in the power diagram, as successively removing users' power intervals. We may therefore similarly visualize successive decoding with time-sharing as successively removing sets of time-power bi-intervals in the time-power diagram.

Now, since individual transmitters are driven by independent data sources, they may not collaborate with each other. However, as observed earlier, a user's transmission in different coherent segments may be decoded together with maximum *a posteriori* decoding to take into account the varying interference from the remaining users' transmissions in different coherent segments. Hence, the receiver may only decode transmissions from a single user during each successive decoding step.

Combining these observations, we have the following conditions on the set of time-power bi-intervals designated for one removal (successive decoding step):

I. all members of the set must belong to a single user, i.e. the set must be a subset of one user's achieving set; and

II. each member of the set must be decodable in the presence of the interference from the remaining users in its respective coherent segment, i.e. the power interval of this member T must be above the the sum of noise variance and the combined power of the remaining users in the coherent segment.

As in straight successive decoding, the number of removal steps required is the number of successive decoding steps.

These observations lead to the following statement of our objective in this chapter:

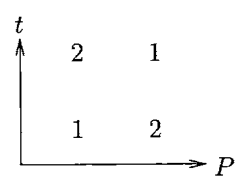
given a set of N-user specifications, construct a time-sharing scheme $\widehat{\mathfrak{U}}$ such that

1. $\widehat{\mathfrak{U}}$ consists of successively decodable coherent segments, and
2. all members time-power bi-intervals of $\widehat{\mathfrak{U}}$ can be removed using no more than $2N-1$ steps overall, each step removing a subset that simultaneously satisfies Conditions I and II above.

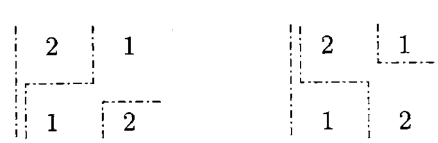
Note that given a time-sharing scheme consisting of successively decodable coherent segments (which satisfies the first objective), the successive decoding order for each coherent segment can be discovered using results in Chapter 3 (c.f. the construction of a p-maximal allocation set). The choice of which user's transmission to decode in the first successive decoding step is limited by such choices within each coherent segment. Moreover, since removing as many time-power bi-intervals as possible during each step is desirable for reducing the total number of steps required, we should therefore remove each time-power bi-interval belonging to the chosen user that can be removed (i.e. above the sum of channel noise intensity and the combined transmission power of the remaining users in their respective coherent segment) in a single step. Finally, noting that the choices of users' transmissions that can be decoded in the next step depends on which user's transmission is decoded in the current step, we conclude that computing the minimal number of removal steps required is a straightforward procedure, although it may require a fairly large number of computations.

We use the two-user case above to illustrate these ideas. As noted above, there are two coherent segments in the time-sharing scheme, the first with decoding order $\{2,1\}$, and the second with decoding order $\{1,2\}$. Since the user to be decoded first in the two coherent segments are different, the first removal step may not remove more than a single time-power bi-interval. Now, observe that regardless of which user's time-power bi-interval is removed in the first step, the second removal step can remove both time-power bi-intervals of the other user. In particular, if $\widehat{T}_{1,2}$ (in the first coherent segment) is removed first, then the second step may remove both time-power bi-intervals in the achieving set of user 1, $\widehat{T}_{1,1}$ and $\widehat{T}_{2,1}$; and if $\widehat{T}_{2,1}$ (in the second coherent segment) is removed first, then the second step may remove both time-power bi-intervals in the achieving set of user 2, $\widehat{T}_{1,2}$ and $\widehat{T}_{2,2}$. Removing the remaining time-power bi-interval at the last step, we conclude that the minimal number of removal steps required for this case is 3.

Note that only the decoding orders of the coherent segments are necessary in the reasoning leading to the minimal number of removal steps required in the two-user case. This makes it possible to consider stacking up the decoding orders for the two coherent segments in the same manner as we stacked the power diagrams for different channel uses to arrive at the following representation The two ways of removing all time-power bi-intervals in the time-sharing scheme



requiring 3 steps can therefore be visualized as follows



(a) (b)

This representation can be easily generalized to be used for computing the required minimal number of removal steps for the arbitrary N-user case. As noted above, no more than N coherent segments are required in a time-sharing scheme to achieve a given set of user specifications. Hence, no more than N decoding orders need to be stacked up in such representations in general.

As a point of curiosity, we consider the case of 3 users. Using the argument in the previous paragraph, we only need to consider groups of three decoding orders. Hence there are $\binom{6}{3} = 20$ such groups. Note that re-numbering the users in all three ordering in a group does not change the required number of removal steps. We may therefore divide these groups into 4 classes. Figure 5-5 illustrates each of these 4 classes using symbols $\{a, b, c\}$. Each member in a class corresponds to one unique one-to-one mapping from $\{a, b, c\}$ to the users' numbers $\{1, 2, 3\}$. Observe that the minimal number of removal steps required for all 4 classes is 5. Since

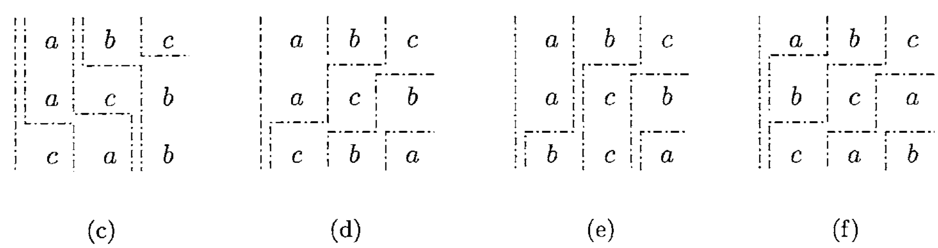


Figure 5-5: Removing all time-power bi-intervals in the 4 classes of time-sharing scheme for 3 users

$2 \times 3 - 1 = 5$, our objective for the 3-user case is accomplished once a time-sharing scheme consisting of 3 coherent segments is found.

Unfortunately, if one chooses an arbitrary set of N decoding orders, each for decoding a set of N users, the minimal number of removal steps required may be greater than $2N - 1$. Figure 5-6 illustrates an example with 4 users which requires a minimal of 8 removal steps.

Our approach to the general problem mimics the one used in treating user-splitting in

a	b	c	d
a	d	c	b
c	a	d	b
d	c	b	a

Figure 5-6: A group of 4 decoding orders for the 4-user case that can not be removed with fewer than $2 \times 4 - 1 = 7$ steps

Chapter 4. This is partly because, in a sense, time-sharing and user-splitting may be seen as dual techniques in achieving successive decoding. In particular, recall that user-splitting makes it possible to decode a virtual user, which has a portion of the power to transmit a part of a user, in each successive decoding step. In other words, both techniques allow a part of a user's message to be decoded in a single successive decoding step. The difference is that time-sharing splits a user's message in time (along the time-axis) while constraining each part of the transmission to observe the average power constraints; whereas user-splitting splits a user's message by splitting the power of transmissions (along the power-axis).

Recall that in Chapter 4, we approached user-splitting by appropriately dividing a set of disjoint power intervals, which is suitable to the given set of user specifications $\mathcal{U}$ (e.g. $\mathsf{irr}(\mathcal{U})$), into a splitting-set of $\mathcal{U}$. The dividing operation was appropriate there because its outcome is a set of disjoint power intervals, which was shown to be successively decodable in Chapter 3.

In this chapter, we will consider an analogous construct on a set of disjoint time-power bi-intervals (which is "suitable" to the given set of user specifications $\mathcal{U}$ for time-sharing). We use the two-user case discussed above to illustrate the ideas the ideas involved.

Let T_1 and T_2 respectively denote the power intervals of these two users. Recall that the specifications of these two users correspond to a rate tuple on the dominant face of the two-user multi-access achievable rate region. From Chapter 2, we know that the combined user is a power interval T which borders on the channel noise intensity n_o. For $i \in \{1, 2\}$, let p_i denote the power specification of user i, i.e.

$$p_i = \mathsf{p}(T_i) \tag{5.15}$$

We have

$$\mathsf{T}(\{T_1, T_2\}) = (n_o, n_o + p_1 + p_2] \tag{5.16}$$

By definition of achieving set, we also know that the combined specification of $\left\{\widehat{T}_{1,1}, \widehat{T}_{2,1}\right\}$ are equal to the specification of T_1, and the combined specification of $\left\{\widehat{T}_{1,2}, \widehat{T}_{2,2}\right\}$ is equal to the specification of T_2. Hence the combined specification of these four time-power bi-intervals must equal the combined specification of T_1 and T_2. In other words, this time-sharing scheme can be seen as a simple *segmentation* of the time-power bi-interval $T \times (0, 1]$, which is the combined user of T_1 and T_2.

Definition 5.1.9. A set of time-power bi-intervals $\widehat{\mathcal{W}}$ is said to be a *segmentation* of a time-power bi-interval $T \times \mathcal{A}$ if members of $\widehat{\mathcal{W}}$ are pair-wise disjoint, and their union is equal to $T \times \mathcal{A}$.

Note that the combined rate contributions of all members of $\widehat{\mathcal{W}}$ is equal to the rate contribution of $T \times \mathcal{A}$.

Note that by definition, the combined rate contribution of all members in a segmentation of $\widehat{\mathfrak{U}}$ is equal to that of all members of $\widehat{\mathfrak{U}}$.

Observe that since the members of a segmentation are pair-wise disjoint, each of the resulting coherent segments must be specified by a set of disjoint power intervals, which is successively decodable from Chapter 3. In other words, we are able to conclude that the resulting time-sharing scheme must be successively decodable without looking into the detailed specifications of its member time-power bi-intervals.

Moreover, since we know, again from Chapter 3, that a set of disjoint power intervals is successively decodable following the descending order of its members, we may proceed to verify whether the resulting time-sharing scheme accomplishes our objective by computing the least number of required removal steps.

To generalize this approach to the arbitrary N-user case, we first generalize the definition of segmentation to operate on sets of pair-wise disjoint time-power bi-intervals.

Definition 5.1.10. Let $\widehat{\mathcal{V}} = \{\widehat{v}_i, i \in [1, \ldots, L]\}$ be a set of pair-wise disjoint time-power bi-intervals. For each $i \in [1, \ldots, L]$, let $\widehat{v}_i$ has a segmentation $\widehat{\mathcal{W}}_i$. Then $\cup_{i=1}^{K} \widehat{\mathcal{W}}_i$ is said to be a *segmentation* of $\widehat{\mathcal{V}}$.

For convenience, we will call the operations leading to a segmentation of a set of time-power bi-intervals the *segmenting* operations.

Following the same reasoning as in the two-user case, we note that every coherent segment in a segmentation of a set of pair-wise disjoint time-power bi-intervals is specified by a set of disjoint power intervals. Hence the segmentation is successively decodable.

In the rest of this chapter, we achieve our objective by constructing time-sharing schemes for a given set of user specifications as appropriate segmentations of the given set's irreducible equivalent. Specifically,

> given a set of N-user specifications $\mathcal{U}$, we explore algorithms that segment $\mathsf{irr}(\mathcal{U})$ into a time-sharing scheme of $\mathcal{U}$ that requires no more than $2N - 1$ removal steps, each simultaneously satisfying Conditions I and II set forth above.

Recall that the irreducible equivalent construction partitions a given set of user specifications into blocks of overlapping subsets. We start with constructing the appropriate time-sharing schemes for overlapping sets in the next section.

5.2 Time-Sharing with Overlapping Sets: Strategy

In the previous section, we extended the power diagram framework into the time-power diagram that contains the appropriate correspondences for studying time-sharing in the AWGN multi-access channels. The extension was accomplished by introducing a new axis called the time-axis. As a result, we are able to use intervals on the time-axis (called time interval) to represent the time portion of the specifications of individual users as well as sets of users. The objective, then, is to achieve successive decoding with time-sharing requiring no more than $2N - 1$ successive decoding steps for an arbitrary set of achievable N-user specifications. The approach is to segment the irreducible equivalent of the given set of user specifications so that our objective can be achieved following Conditions I and II above.

As in our treatment of the simplification of the AWGN achievability and successive decoding with user splitting, the generalization to arbitrary irreducible equivalents turns out to be straightforward once the case of overlapping sets (which have a singleton irreducible equivalent) is well-understood. Starting in this section, we consider time-sharing for overlapping sets of user specifications.

As we will see, the construction algorithms introduced here generally have more complex descriptions than those introduced in previous chapters. Part of the reason for the additional complexity is the fact that they deal with two-dimensional entities, (time-power bi-intervals) instead of single-dimensional entities (i.e. power-intervals). However, because the representation in the time-power diagram encapsulates all the characteristic and specifications in time-sharing compactly, the intuitions behind the design of these algorithms are straight-forward. In this section, we start with presenting these intuitions.

Our algorithms follow a similar approach as the set of conservation algorithms presented for user-splitting in Chapter 4. Specifically, each iteration of an algorithm starts with choosing a user from the remaining users (those without assignments of achieving sets). Then, an achieving set for the chosen user is constructed following the *conservation principle*, i.e. as the left-over of some appropriately constructed set of time-power bi-intervals.

The progression of the algorithms resemble that of the simple algorithm for user-splitting, Algorithm 4.2.3 (rather, they resemble the conservation algorithm that allocates the irreducible equivalent for the remaining users). Specifically, each iteration of an algorithm partitions the remaining users into increasingly smaller blocks. In the simple algorithm, each small block represents a largest overlapping subset in the set of remaining users (excluding the chosen one). As we will see, each block constructed in the algorithm here also represents a largest subset in the set of remaining users (again excluding the chosen one) satisfying certain conditions that generalizes the overlapping conditions.

We use the first iteration of our algorithm to illustrate these ideas.

Let $\mathcal{U} = \{u_i, i \in [1, \dots, N]\}$ be the given set of user power intervals with the overlapping property. Let $\mathsf{T}(\mathcal{U}) = (a, b]$.

Without the loss of generality, let u_1 be the first chosen user. Let $\mathcal{V} = \mathcal{U} \setminus \{u_1\}$. Let L_1 denote the number of power intervals in $\mathsf{irr}(\mathcal{V})$. Recall that $[\mathsf{irr}(\mathcal{V})]_l < [\mathsf{irr}(\mathcal{V})]_{l+1}$ for all feasible l.

For convenience, we define the following notation.

Definition 5.2.1. Given an ordered set of power intervals $\mathcal{W} = \{w_j, j \in [1, \dots, J]\}$. For each $j \in [1, \dots, J]$, let $\Sigma^f_{\mathcal{W}}(j)$ denote the combined power specifications of the first through j^{th} member of $\mathcal{W}$, i.e.

$$\Sigma^f_{\mathcal{W}}(j) = \sum_{j'=1}^{j} \mathsf{p}\left(w_{j'}\right) \tag{5.17}$$

Additionally, define $\Sigma^f_{\mathcal{W}}(0) = 0$.

Similarly, let $\Sigma^b_{\mathcal{W}}(j)$ denote the combined power specifications of the j^{th} through the last member of $\mathcal{W}$, i.e.

$$\Sigma^b_{\mathcal{W}}(j) = \sum_{j'=j}^{J} \mathsf{p}\left(w_{j'}\right) \tag{5.18}$$

Additionally, define $\Sigma^b_{\mathcal{W}}(J+1) = 0$.

Note that, for each $j \in [0, \dots, J]$,

$$\Sigma^f_{\mathcal{W}}(j) + \Sigma^b_{\mathcal{W}}(j+1) = \sum_{j'=1}^{J} \mathsf{p}\left(w_{j'}\right) = \mathsf{p}\left(\mathsf{T}(\mathcal{W})\right) \tag{5.19}$$

The first iteration of our algorithm consists of three sets of operations. We start with the first set.

Procedure 5.2.1 (The 1^{th} set of operations for the first iteration).
For each $l \in [1, \dots, L_1]$, let

$$v_{l,1} = \left(a + \Sigma^f_{\mathsf{irr}(\mathcal{V})}(l-1), a + \Sigma^f_{\mathsf{irr}(\mathcal{V})}(l)\right] \tag{5.20}$$

$$v_{l,2} = \left(b - \Sigma^b_{\mathsf{irr}(\mathcal{V})}(l), b - \Sigma^b_{\mathsf{irr}(\mathcal{V})}(l+1)\right] \tag{5.21}$$

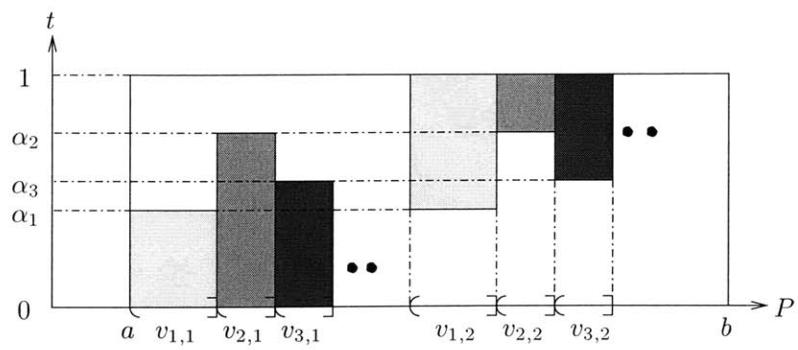


Figure 5-7: The construction outcome of the first set of operations in the first iteration

and let $\alpha_l \in [0,1)$ *be such that the set of two time-power bi-intervals*

$$\widehat{\mathcal{V}}_l = \{v_{l,1} \times (0, \alpha_l], v_{l,2} \times (\alpha_l, 1]\} \tag{5.22}$$

is an achieving set of $[\mathsf{irr}(\mathcal{V})]_l$.

In Figure 5-7, we use different shades of darkness to distinguish the constructed achieving sets for different users.

Note that for each $j \in \{1,2\}$, $v_{l,j}$ and $v_{l+1,j}$ are adjacent with $v_{l,j} < v_{l+1,j}$. Hence $\{v_{l,j}, l \in [1,\ldots,L_1\}$ is an adjacent set for each j.

To show that the desired $\widehat{\mathcal{V}}_l$ exists for each l, we start with observing that

$$\mathsf{p}(v_{l,1}) = \left(a + \Sigma^f_{\mathsf{irr}(\mathcal{V})}(l)\right) - \left(a + \Sigma^f_{\mathsf{irr}(\mathcal{V})}(l-1)\right) = \mathsf{p}([\mathsf{irr}(\mathcal{V})]_l) \tag{5.23}$$

$$\mathsf{p}(v_{l,2}) = \left(a + \Sigma^b_{\mathsf{irr}(\mathcal{V})}(l)\right) - \left(a + \Sigma^b_{\mathsf{irr}(\mathcal{V})}(l+1)\right) = \mathsf{p}([\mathsf{irr}(\mathcal{V})]_l) \tag{5.24}$$

Hence,

$$\mathsf{p}(v_{l,1}) = \mathsf{p}(v_{l,2}) \tag{5.25}$$

Let $[\mathsf{irr}(\mathcal{V})]_l = (a_l, b_l]$. Since $\mathcal{V} \subset \mathcal{U}$ and $\mathcal{U}$ is an overlapping set, we have

$$\mathsf{ext}(a_1, b_{L_1}] = (\mathsf{irr}(\mathcal{V})) \subseteq \mathsf{T}(\mathcal{U}) = (a, b]$$

Combined with the fact that $[\mathsf{irr}(\mathcal{V})]_l < [\mathsf{irr}(\mathcal{V})]_{l+1}$, we have

$$a + \Sigma^f_{\mathsf{irr}(\mathcal{V})}(l-1) \le a_l < b_l \le b - \Sigma^b_{\mathsf{irr}(\mathcal{V})}(l+1) \tag{5.26}$$

By the monotonicity of the $\mathsf{r}(\cdot)$ function, we therefore have

$$\mathsf{r}(v_{l,1}) \leq \mathsf{r}([\mathsf{irr}(\mathcal{V})]_l) \leq \mathsf{r}(v_{l,2}) \tag{5.27}$$

This confirms that α_l is in $[0,1)$ for each l, and thus asserts that the first set of operations complete successfully.

Before continuing our discussion, let me point out that the lower boundary of $v_{l,2}$, $b - \Sigma^b_{\mathsf{irr}(\mathcal{V})}(l)$, may not be greater than the upper boundary of $v_{l,1}$, $a + \Sigma^f_{\mathsf{irr}(\mathcal{V})}(l)$ in general. We have drawn it this way in Figure 5-7 only to simplify the illustration.

Note at the end of this set of operations, that α_l's do not necessarily follow any order. This fact can be shown by Theorem F.0.8 in Appendix F. In the next set operations, we will process the outcomes of the first set of operations so that these parameters will follow a descending order.

The second set of operations (of the first iteration) progresses in a manner that is similar to the algorithm that constructs the irreducible equivalent. In particular, every step of this set of operations compares two adjacent α_l's. If $\alpha_l \leq \alpha_{l+1}$, the achieving set for the combined user of the two members in the irreducible equivalent is constructed, and two original achieving sets are replaced by the newly constructed one. We detail this set of operations below.

Procedure 5.2.2 (The 2^{nd} set of operations for the first iteration).
For convenience, we make the following assignments. For each $l \in [1, \ldots, L_1]$, we let $\lambda_l = \alpha_l$, $S_l = [\mathsf{irr}(\mathcal{V})]_l$ and $\mathcal{V}_l = \mathsf{B}_l(\mathcal{V})$. Let $\mathcal{W} = \{S_l, l \in [1, \ldots, L_1]\}$. Initialize $l = 1$. Perform the following.

Step 1. *If S_l is the last element in $\mathcal{W}$, exit the algorithm. Otherwise*

Step 2. *If $\lambda_l > \lambda_{l+1}$ are disjoint, then increase l by 1 and go to Step 2.*

Step 3. *Otherwise $\lambda_l \leq \lambda_{l+1}$. Perform the following:*

- *(a) replace S_l and S_{l+1} by their combined user $\mathsf{T}(\{S_l, S_{l+1}\})$, i.e. let $S_l = \mathsf{T}(\{S_l, S_{l+1}\})$*
- *(b) replace $\mathcal{V}_l$ and $\mathcal{V}_{l+1}$ by their union, i.e. let $\mathcal{V}_l = \mathcal{V}_l \cup \mathcal{V}_{l+1}$*
- *(c) for each $l' > l$ starting from $l' = l+1$, update $\mathcal{W}$ by renaming $S_{l'} = S_{l'+1}$, $\mathcal{V}_{l'} = \mathcal{V}_{l'+1}$, and $\lambda_{l'} = \lambda_{l'+1}$ (and reduce the number of S_l, $\mathcal{V}'_l$ and λ_l by 1)*

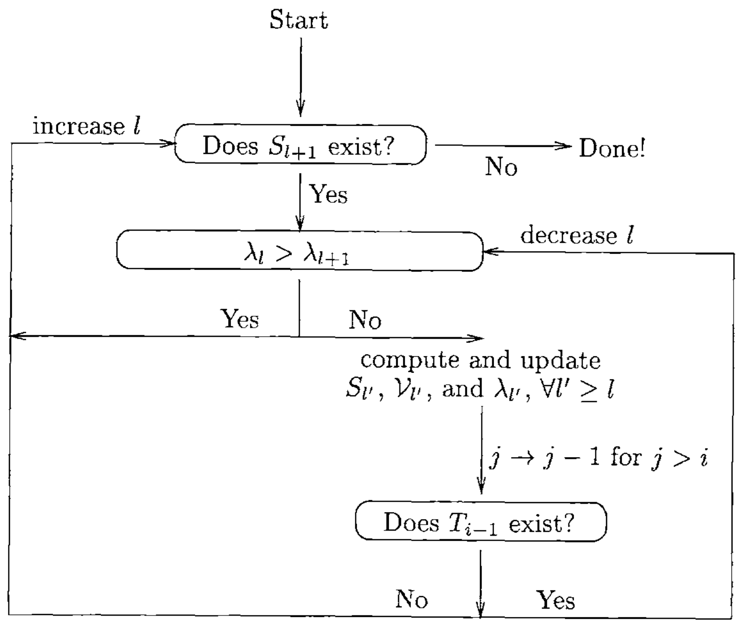


Figure 5-8: Flowchart of the second set of operations in the first iteration as in Procedure 5.2.2

(d) let

$$w_{l,1} = \left(a + \Sigma^f_{\mathcal{W}}(l-1), a + \Sigma^f_{\mathcal{W}}(l)\right] \tag{5.28}$$

$$w_{l,2} = \left(b - \Sigma^b_{\mathcal{W}}(l), b - \Sigma^b_{\mathcal{W}}(l+1)\right] \tag{5.29}$$

and compute the new λ_l such that

$$\widehat{\mathcal{W}}_l = \{w_{l,1} \times (0, \lambda_l], w_{l,2} \times (\lambda_l, 1]\} \tag{5.30}$$

is an achieving set for the updated S_l

If S_{l-1} exists, decrease l by 1, and go to Step 2; else go to Step 1

Let L_2 denote the number of power intervals in $\mathcal{W}$, which is the value of l at the end of the computation.

Figure 5-8 presents a flowchart of this set of operations. Note the progression of this algorithm is identical to that of Algorithm 2.3.8 in Figure 2-11.

To understand this algorithm, first note that $\mathcal{W}$ is initialized to be the disjoint set of power intervals $\{S_l, l \in [1, \ldots, L_1]\}$. Since $S_l = [\mathsf{irr}(\mathcal{V})]_l < [\mathsf{irr}(\mathcal{V})]_{l+1} = S_{l+1}$, from Theorem 2.2.1, we have $\mathsf{T}(\{S_l, S_{l+1}\}) \subseteq \mathsf{ext}(\{S_l, S_{l+1}\})$. Hence the new $\mathcal{W}$ after the update is again a set of disjoint power intervals with $S_l < S_{l+1}$.

Additionally, note that $\mathcal{V}_l$ is initialized to be $\mathsf{B}_l(\mathcal{V})$, and that $\{\mathcal{V}_l, l \in [1, \ldots, L_1]\}$ forms a partition of $\mathcal{V}$. Combined with the fact that S_l is initialized to be $\mathsf{T}(\mathcal{V}_l)$ at the beginning of this set of operations, we conclude that, when this set of operations are completed,

$$S_l = \mathsf{T}(\mathcal{V}_l) \tag{5.31}$$

and the resulting set of $\mathcal{V}_l$'s remains to be a partition of V.

Also note that for each $j \in \{1, 2\}$, $w_{l,j}$ and $w_{l+1,j}$ are adjacent with $w_{l,j} < w_{l+1,j}$. Hence $\{w_{l,j}, l \in [1, \dots, L_1\}$ is an adjacent set for each j. (Note that the arguments used here are identical to those used to show that $\{\widehat{v}_{l,j}, l \in [1, \dots, L_1]\}$ is an adjacent set for each j.)

The objective of this set of operations is to achieve operations,

$$\lambda_l > \lambda_{l+1}, \quad \forall l \in [1, \dots, L_2] \tag{5.32}$$

Figure 5-9 illustrates the achieving sets, $\widehat{\mathcal{W}}_l$'s, constructed for members of (the resulting) $\mathcal{V}'$.

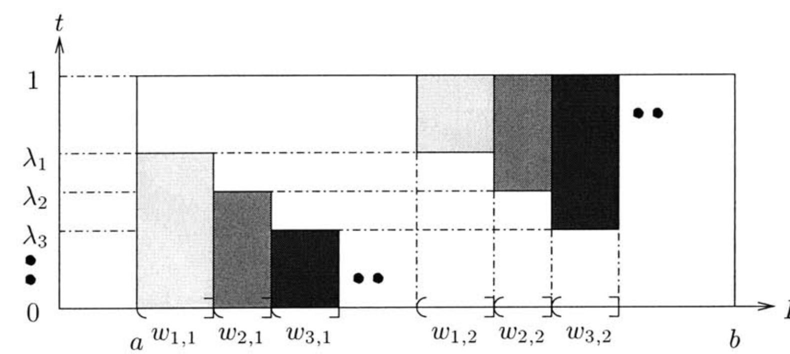


Figure 5-9: The construction outcome of the second set of operations in the first iteration

To understand (5.32), we compare the set of λ_l's before and after the computations (and updates) in Step 3 for a particular l (assuming previous iterations have resulted in $\lambda_{l'}$'s satisfying (5.32) for all $l' < l - 1$).

For notational convenience, let us save this set of values before Step 3 in the set of λ'_l's. Since Step 3 is to be performed, we know that for this particular l, we have

$$\lambda'_l \leq \lambda'_{l+1} \tag{5.33}$$

Also, we save the values of $w_{l,i}$ before Step 3 as $w'_{l,i}$ for each $i \in \{1, 2\}$, save $\widehat{\mathcal{W}}_l$ before this step as $\widehat{\mathcal{W}}'_l$, and save S_l before this step as S'_l.

Note that, since all S'_l's for $l' < l$ and for $l + 1 < l'$ are not affected by this iteration (except

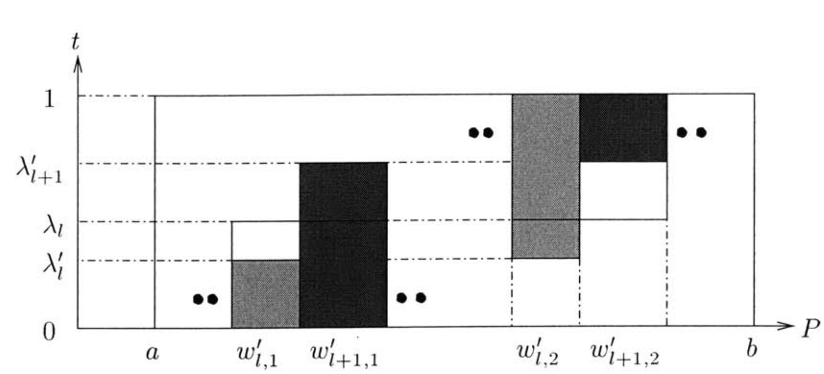


Figure 5-10: Illustrating terms used in (5.37) as time-power bi-intervals

the index of those with $l+1<l'$ are shifted down by 1), we have

$$w_{l,i} = w'_{l,j} \cup w'_{l+1,j}, \quad \forall j \in \{1,2\} \tag{5.34}$$

Since, from the previous computations, $\widehat{\mathcal{W}}'_l = \left\{w'_{l,1} \times (0,\lambda'_l], w'_{l,2} \times (\lambda'_l, 1]\right\}$ and $\widehat{\mathcal{W}}'_{l+1} = \left\{w'_{l+1,1} \times \left(0,\lambda'_{l+1}\right], w'_{l+1,2} \times \left(\lambda'_{l+1}, 1\right]\right\}$ are achieving sets for S'_l and S'_{l+1} respectively, we have

$$\mathsf{r}\left(S'_l\right) = \bar{\mathsf{r}}\left(\widehat{\mathcal{W}}'_l\right) = \lambda'_l \mathsf{r}\left(w'_{l,1}\right) + \left(1-\lambda'_l\right)\mathsf{r}\left(w'_{l,2}\right) \tag{5.35}$$

$$\mathsf{r}\left(S'_{l+1}\right) = \bar{\mathsf{r}}\left(\widehat{\mathcal{W}}'_{l+1}\right) = \lambda'_{l+1} \mathsf{r}\left(w'_{l+1,1}\right) + \left(1-\lambda'_{l+1}\right)\mathsf{r}\left(w'_{l+1,2}\right) \tag{5.36}$$

Summing the two sides of these equations, and after some manipulation using (5.33), we have

$$\begin{aligned} \mathsf{r}\left(S'_l\right) + \mathsf{r}\left(S'_{l+1}\right) &= \lambda'_l \mathsf{r}\left(w'_{l,1} \cup w'_{l+1,1}\right) + \left(1-\lambda'_{l+1}\right)\mathsf{r}\left(w'_{l,2} \cup w'_{l+1,2}\right) + \\ &\quad + \left(\lambda'_{l+1} - \lambda'_l\right)\left(\mathsf{r}\left(w'_{l+1,1}\right) + \mathsf{r}\left(w'_{l,2}\right)\right) \\ &= \lambda'_l \mathsf{r}\left(w_{l,1}\right) + \left(1-\lambda'_l\right)\mathsf{r}\left(w_{l,2}\right) + \left(\lambda'_{l+1} - \lambda'_l\right)\left(\mathsf{r}\left(w'_{l+1,1}\right) + \mathsf{r}\left(w'_{l,2}\right)\right) \end{aligned} \tag{5.37}$$

Figure 5-10 illustrates the terms used in the discussion here as time-power bi-intervals.

Let $\mathcal{W}'$ denote the set $\mathcal{W}$ before Step 3. Recall that $\Sigma^f_{\mathcal{W}'}(l-1) + \Sigma^b_{\mathcal{W}'}(l)$ is the combined power specifications of all power intervals in $\mathcal{W}'$. Since each member of $\mathcal{W}'$ is the combined user of some distinct subset of $\mathcal{V}$, and $\mathcal{V} \subseteq \mathcal{U}$, we conclude that

$$\Sigma^f_{\mathcal{W}'}(l-1) + \Sigma^b_{\mathcal{W}'}(l) < \mathsf{p}(\mathsf{T}(\mathcal{U})) = b - a$$

In other words, we have, for all feasible l

$$a + \Sigma^f_{\mathcal{W}'}(l-1) < b - \Sigma^b_{\mathcal{W}'}(l) \tag{5.38}$$

By the monotonicity of $\mathsf{r}(\cdot)$ function, we have, for all feasible l,

$$\mathsf{r}\left(w'_{l,1}\right) > \mathsf{r}\left(w'_{l,2}\right) \tag{5.39}$$

We therefore have

$$\mathsf{r}\left(w'_{l,1}\right) + \mathsf{r}\left(w'_{l+1,1}\right) > \mathsf{r}\left(w'_{l+1,1}\right) + \mathsf{r}\left(w'_{l,2}\right) > \mathsf{r}\left(w'_{l,2}\right) + \mathsf{r}\left(w'_{l+1,2}\right) \tag{5.40}$$

Note the adjacency of $w'_{l,1}$ and $w'_{l+1,1}$, and the adjacency of $w'_{l,2}$ and $w'_{l+1,2}$, we have

$$\mathsf{r}\left(w_{l,1}\right) = \mathsf{r}\left(w'_{l,1} \cup w'_{l+1,1}\right) > \mathsf{r}\left(w'_{l+1,1}\right) + \mathsf{r}\left(w'_{l,2}\right) > \mathsf{r}\left(w'_{l,2} \cup w'_{l+1,2}\right) = \mathsf{r}\left(w_{l,2}\right) \tag{5.41}$$

Multiplying all sides by $\left(\lambda'_{l+1} - \lambda'_l\right)$, and adding

$$\lambda'_l \mathsf{r}\left(w'_{l,1} \cup w'_{l+1,1}\right) + \left(1 - \lambda'_{l+1}\right) \mathsf{r}\left(w'_{l,2} \cup w'_{l+1,2}\right) = \lambda'_l \mathsf{r}\left(w_{l,1}\right) + \left(1 - \lambda'_{l+1}\right) \mathsf{r}\left(w_{l,2}\right)$$

we have

$$\begin{aligned} \lambda'_{l+1} \mathsf{r}\left(w_{l,1}\right) + \left(1 - \lambda'_{l+1}\right) \mathsf{r}\left(w_{l,2}\right) \; &> \; \mathsf{r}\left(S'_l\right) + \mathsf{r}\left(S'_{l+1}\right) > \\ &> \; \lambda'_l \mathsf{r}\left(w_{l,1}\right) + \left(1 - \lambda'_l\right) \mathsf{r}\left(w_{l,2}\right) \end{aligned} \tag{5.42}$$

Finally, since $S_l = \mathsf{T}\left(\left\{\mathsf{r}\left(S'_l\right) + \mathsf{r}\left(S'_{l+1}\right)\right\}\right)$, we have

$$\lambda'_l < \lambda_l < \lambda'_{l+1} \tag{5.43}$$

Note that if $\lambda'_{l-1} > \lambda'_l < \lambda'_{l+1}$, it is possible to have $\lambda'_{l-1} \leq \lambda_l$ after this computation. This case is covered at the end of Step 3. We have thus established (5.32) to be the outcome of this set of operations.

Before presenting the 3^{th} (and the last) set of operations, we extend the following definitions from Chapter 4.

Definition 5.2.2. Let sets $\widehat{\mathcal{V}}$ and $\widehat{\mathcal{W}}$ each consist of pair-wise disjoint time-power bi-intervals. Then, we say that $\widehat{\mathcal{W}}$ *t-precedes* $\widehat{\mathcal{V}}$ *(or* $\widehat{\mathcal{V}}$ *t-succeeds* $\widehat{\mathcal{W}}$*)* if every member time-power bi-interval in $\widehat{\mathcal{W}}$ is contained in a member time-power bi-interval in $\widehat{\mathcal{V}}$, denote it by $\widehat{\mathcal{W}} \prec \widehat{\mathcal{V}}$ (or $\widehat{\mathcal{V}} \succ \widehat{\mathcal{W}}$).

Observe that every achieving set constructed in the first two sets of operations t-precedes $\mathsf{T}(\mathcal{U}) \times (0,1] = (a,b] \times (0,1]$.

Definition 5.2.3. Let $\widehat{\mathcal{V}}$ and $\widehat{\mathcal{W}}$ each denote a set of pair-wise disjoint time-power bi-intervals.

Suppose $\widehat{\mathcal{W}} \prec \widehat{\mathcal{V}}$. Let $\widehat{\mathcal{V}}'$ be a segmentation of $\widehat{\mathcal{V}}$ consisting of the least number of time-power bi-intervals such that each time-power bi-interval in $\widehat{\mathcal{W}}$ is a time-power bi-interval in $\widehat{\mathcal{V}}'$, i.e. $\widehat{\mathcal{W}} \subseteq \widehat{\mathcal{V}}'$. Then $\widehat{\mathcal{V}}' \setminus \widehat{\mathcal{W}}$ is said to be the *t-complementing set* of $\widehat{\mathcal{W}}$ in $\widehat{\mathcal{V}}$, denote it by $\widehat{\mathsf{cmpl}}(\widehat{\mathcal{V}}, \widehat{\mathcal{W}})$

Observe that, in Figures 5-7 through 5-10, the un-shaded areas inside $(a, b] \times (0, 1]$ is the t-complementing set of the union of the achieving sets constructed (the shaded rectangles).

In the following, we will show that the remaining un-shaded areas form an achieving set for u_1 – the user we have excluded from $\mathcal{V}$.

To see this, we start with the case when $L_2 = 2$, i.e. the second set of operations end with $l = 2$. We know that all users in $\mathcal{V}$ are partitioned into 2 blocks, $\mathcal{V}_1$ and $\mathcal{V}_2$. Figure 5-11 illustrates this case.

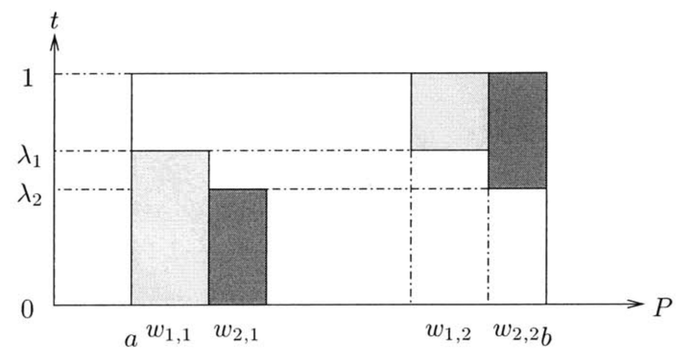


Figure 5-11: The construction outcome of the second set of operations (in the first iteration) when the operations end with $L_2 = 2$

Let

$$u_{1,1} = \left(a + \Sigma^{f}_{\widehat{\mathcal{W}}}(2), b - \Sigma^{b}_{\widehat{\mathcal{W}}}(3)\right] \tag{5.44}$$

$$u_{1,2} = \left(a + \Sigma^{f}_{\widehat{\mathcal{W}}}(1), b - \Sigma^{b}_{\widehat{\mathcal{W}}}(2)\right] \tag{5.45}$$

$$u_{1,3} = \left(a, b - \Sigma^{b}_{\widehat{\mathcal{W}}}(1)\right] \tag{5.46}$$

Note that these un-shaded areas may be seen as the union of three time-power bi-intervals:

$$u_{1,1} \times (0, \lambda_2], u_{1,2} \times (\lambda_2, \lambda_1], u_{1,3} \times (\lambda_1, 1]$$

Notice that all three have identical power specifications

$$b - a - \sum_{i=2}^{N} \mathsf{p}(u_i)$$

Since $\mathcal{V} = \mathcal{U} \setminus \{u_1\}$, we have

$$\mathsf{p}(\mathsf{T}(\mathcal{U})) = b - a = \mathsf{p}(u_1) + \sum_{i=2}^{N} \mathsf{p}(u_i) \tag{5.47}$$

It is therefore clear that these three time-power bi-intervals have the same power specifications as u_1.

Moreover, recall, from the definition of segmentation, the combined rate contribution of all members in a segmentation of $(a, b]$ is equal to the rate specification of $(a, b]$. Since $\mathsf{r}((a, b]) = \mathsf{r}(\mathsf{T}(\mathcal{U}))$, we conclude that the combined rate contribution of the three time-power bi-intervals above (the unshaded area) must be the same as the rate specification of u_1. In other words, these three time-power bi-intervals form an achieving set for u_1.

The following theorem generalizes this result.

Theorem 5.2.3. *Let T_1 and T_2 denote two power intervals, and let $\widehat{\mathcal{W}}$ be an achieving set of $\mathsf{T}(\{T_1, T_2\})$. Let $\widehat{\mathcal{W}}_1 \cup \widehat{\mathcal{W}}_2$ be a segmentation of $\widehat{\mathcal{W}}$. Suppose, in addition, that $\widehat{\mathcal{W}}_1$ and $\widehat{\mathcal{W}}_2$ are two sets of time-power bi-intervals with the simple property such that $\widehat{\mathcal{W}}_1$ is an achieving set of T_1. Then the $\widehat{\mathcal{W}}_2$ must be an achieving set for T_2.*

Remark. Recall that in our study of user-splitting, Theorem 4.2.1 established that the complementing set, after choosing a suitable set for the remaining users, is a splitting-set for the chosen user in each iteration of the conservation algorithm. Hence, the current theorem can be seen as its generalization into the time-power diagram framework. Finally, since the result of the current theorem is a simple consequence of the definitions of time-power bi-intervals and achieving sets. We omit its proof for simplicity.

We have the 3rd (and the final) set of operations for this case.

Procedure 5.2.4 (The 3rd (and the last) set of operations for the first iteration). *Construct $\widehat{\mathsf{cmpl}}\left(\mathsf{T}(\mathcal{U}), \bigcup_{l=1}^{L_2} \widehat{\mathcal{W}}_l\right)$, and assign it to be the achieving set of u_1.*

Note that since $\widehat{\mathcal{W}}_1$ and $\widehat{\mathcal{W}}_2$ are achieving sets of S_1 and S_2 respectively, $\widehat{\mathcal{W}}_1$, $\widehat{\mathcal{W}}_2$ and the three time-power bi-intervals form a time-sharing scheme for $\{u_1, S_1, S_2\}$. Moreover, notice that we may remove this time-sharing scheme (observing Conditions I and II) using the following order:

1. remove $u_{1,1} \times (0, \lambda_2]$

2. remove $\widehat{\mathcal{W}}_2$

3. remove $u_{1,2} \times (\lambda_2, \lambda_1]$

4. remove $\widehat{\mathcal{W}}_1$

5. remove $u_{1,3} \times (\lambda_1, 1]$

We therefore conclude that this time-sharing scheme can be removed (decoded) in 5 steps.

In general, this method of removing a member in the achieving set of u_1 followed by removing one of the $\widehat{\mathcal{W}}_l$'s in its entirety can be used to remove the construction outcome of this first iteration with arbitrary L_2. In particular, notice that such a removal scheme requires no more than $2L_2 + 1$ steps.

We now use this case of $L_2 = 2$ to illustrate the strategy to proceed in our construction.

For each $l \in \{1, 2\}$, let m_j denote the number of power intervals (users) in $\mathcal{V}_l$. Note that this implies that the total number of users in the original specification $\mathcal{U}$ is $m_1 + m_2 + 1$, i.e.

$$N = m_1 + m_2 + 1$$

Observe that, if we could segment $\widehat{\mathcal{W}}_l$ into an achieving set of $\mathcal{V}_l$ such that no more than $2m_l - 1$ decoding steps (removal steps) are required to decode all time-power bi-intervals in the resulting segmentation (following Conditions I and II above), we may then modify the decoding order for the time-sharing scheme of $\{u_1, S_1, S_2\}$ above so that the decoding order for the segmentation of $\widehat{\mathcal{W}}_l$ is followed in the step that removes $\widehat{\mathcal{W}}_l$. Specifically,

1. remove $u_{1,1} \times (0, \lambda_2]$

2. remove the segmentation of $\widehat{\mathcal{W}}_2$ following its decoding order

3. remove $u_{1,2} \times (\lambda_2, \lambda_1]$

4. remove the segmentation of $\widehat{\mathcal{W}}_1$ following its decoding order

5. remove $u_{1,3} \times (\lambda_1, 1]$

Note that the total number of removal steps required is

$$3 + (2m_1 - 1) + (2m_2 - 1) = 2(m_1 + m_2 + 1) - 1 = 2N - 1$$

This would have accomplished our objective.

Following this strategy, all that's left to be done is to find a method of constructing the desired segmentation of $\widehat{\mathcal{W}}_l$. Note that the only difference between segmenting $\widehat{\mathcal{W}}_l$ into a desirable time-sharing scheme for $\mathcal{V}_l$ and our original goal, which is to segment $\mathsf{T}(\mathcal{U})$ into a desirable time-sharing scheme for $\mathcal{U}$, is that $\widehat{\mathcal{W}}_j$ is a simple set of time-power bi-intervals, and $\mathsf{T}(\mathcal{U})$ is a single time-power bi-interval.

In Appendix E, we show that a simple set of time-power bi-intervals is equivalent to a generalized power interval (in an appropriate generalize power diagram), where the generalized power interval is a generalization of the power interval representation to take into account of time-sharing. Moreover, using the following definitions, we have established Theorem 5.2.5 and 5.2.6 in Appendix G.

Definition 5.2.4. Given a power interval $T = (a, a+p]$, and $0 \leq \delta \leq p$. Let $\mathsf{G}(T,\delta)$ $(\mathsf{L}(T,\delta))$ denote the power interval with power constraint δ that is the member of a division of $\widehat{T}$ that achieves the maximal (minimal) rate.

By the monotonicity of $\mathsf{r}(\cdot)$ function, note that

$$\mathsf{G}(T,\delta) = (a, a+\delta] \tag{5.48}$$

$$\mathsf{L}(T,\delta) = (a+p-\delta, a+p] \tag{5.49}$$

Also

$$\mathsf{G}(T,\delta) = (a, a+\delta] = \mathsf{cmpl}\left(T, \mathsf{L}\left(T, p-\delta\right)\right) \tag{5.50}$$

Definition 5.2.5. Given a time-power bi-interval $\widehat{T}$, and $0 < \delta \leq \mathsf{p}\left(\widehat{T}\right)$. Let $\widehat{\mathsf{G}}\left(\widehat{T},\delta\right)$ $(\widehat{\mathsf{L}}\left(\widehat{T},\delta\right))$ denote the time-power bi-interval with power specification δ and time interval $\widehat{\mathsf{I}}_t\left(\widehat{T}\right)$ that is the member of a division of $\widehat{T}$ that achieves the maximal (minimal) average transmission rate.

Observe that

$$\widehat{\mathsf{G}}\left(\widehat{T},\delta\right) = \mathsf{G}\left(\widehat{\mathsf{I}}_p\left(\widehat{T}\right),\delta\right) \times \widehat{\mathsf{I}}_t(\widehat{T}) \tag{5.51}$$

$$\widehat{\mathsf{L}}\left(\widehat{T},\delta\right) = \mathsf{L}\left(\widehat{\mathsf{I}}_p\left(\widehat{T}\right),\delta\right) \times \widehat{\mathsf{I}}_t(\widehat{T}) \tag{5.52}$$

Definition 5.2.6. Given a set of simple time-power bi-intervals $\widehat{\mathcal{V}}$, and $0 < \delta < \overline{\mathsf{p}}\left(\widehat{\mathcal{V}}\right)$. Let $\widehat{\mathsf{G}}\left(\widehat{\mathcal{V}},\delta\right)$ $(\widehat{\mathsf{L}}\left(\widehat{\mathcal{V}},\delta\right))$ denote the simple set of time-power bi-intervals with power specification δ that is the subset of a segmentation of $\widehat{\mathcal{V}}$ that achieves the maximal (minimal) rate specification.

Theorem 5.2.5. *Let $\mathcal{V} = \{T_i, i \in [1..m]\}$ be a set of disjoint power intervals with $T_i < T_{i+1}$.*

Let $\widehat{\mathcal{W}}$ be an achieving set of $\mathsf{T}(\mathcal{V})$. *Suppose*

$$\sum_{j=1}^{i} \mathsf{r}(T_j) \leq \bar{\mathsf{r}}\left(\widehat{\mathsf{G}}\left(\widehat{\mathcal{W}}, \sum_{j=1}^{i} \mathsf{p}(T_j)\right)\right), \quad \forall i \in [1..m] \tag{5.53}$$

Then for all $\delta \in [0, \mathsf{p}(\mathsf{T}(\mathcal{V}))]$,

$$\mathsf{r}(\mathsf{G}(\mathcal{V}, \delta)) \leq \bar{\mathsf{r}}\left(\widehat{\mathsf{G}}\left(\widehat{\mathcal{W}}, \delta\right)\right) \tag{5.54}$$

Theorem 5.2.6. *Let $\mathcal{V} = \{T_i, i \in [1..m]\}$ be a set of disjoint power intervals with $T_i < T_{i+1}$. Let $\widehat{\mathcal{W}}$ be an achieving set of* $\mathsf{T}(\mathcal{V})$ *with*

$$\sum_{j=i}^{M} \mathsf{r}(T_j) \geq \bar{\mathsf{r}}\left(\widehat{\mathsf{L}}\left(\widehat{\mathcal{W}}, \sum_{j=i}^{M} \mathsf{p}(T_j)\right)\right), \quad \forall i \in [1..M] \tag{5.55}$$

Then for all $\delta \in [0, \mathsf{p}(\mathsf{T}(\mathcal{V}))]$,

$$\mathsf{r}(\mathsf{L}(\mathcal{V}, \delta)) \geq \bar{\mathsf{r}}\left(\widehat{\mathsf{L}}\left(\widehat{\mathcal{W}}, \delta\right)\right) \tag{5.56}$$

These two theorems can be used in conjunction with properties of overlapping set to establish that the block of users $\mathcal{V}_l$ is in fact an overlapping set in the generalized power diagram (the one in which $\widehat{\mathcal{W}}_l$ is represented as a single power interval). Therefore, we may repeat the approach presented above to segment $\widehat{\mathcal{W}}_l$ using the generalized power diagram.

Specifically, we extend the generalized power diagram to generalized time-power diagram, and repeat the 3 sets of operations presented above to accomplish the following:

- choose a power interval $T \in \mathcal{V}_l$, construct $\mathcal{V}'_l = \mathcal{V}_j \setminus \{T\}$
- partition $\mathcal{V}'_l$ into L_2 blocks, such that the combined user of each block has $\widehat{\mathcal{W}}_l$ as achieving set, and all the achieving sets follow identical structure (in the generalized time-power diagram) as the $\widehat{\mathcal{W}}_l$'s in Figure 5-9

Note that the resulting achieving sets are time-sharing of the time-shared results from the first iteration, which is no different from a simple set of time-power bi-intervals. Figure 5-12 illustrates the construction outcome of this iteration using the time-power diagram. We again use difference in shading to distinguish the achieving sets constructed for the resulting blocks.

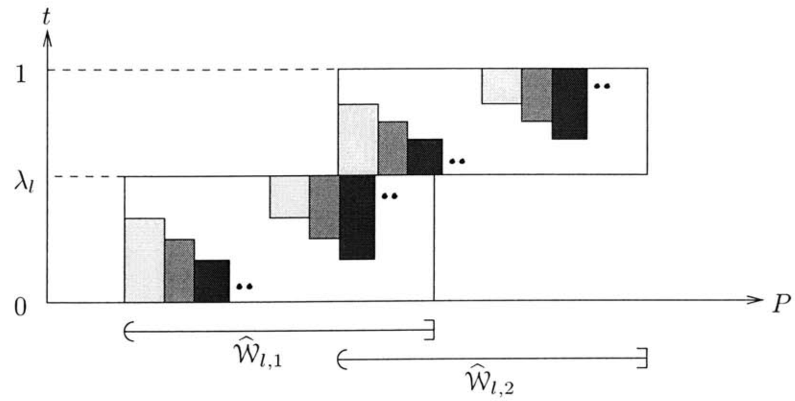


Figure 5-12: The construction outcome of the second set of operations in the second iteration

Finally, we repeat this iteration for each resulting block that contains more than a single user. When the iteration ends, we have constructed the desired time-sharing scheme[6]

Before leaving this chapter, we note that it is also possible construct the achieving sets in the second iteration using constructions illustrated in Figure 5-13. Using the least shaded set of time-power bi-intervals as an example, this is because when $\lambda_{l,1}$ takes on values 0 or 1, the combined rate contribution of the set of time-power bi-intervals is equal to the corresponding set of time-power bi-intervals in Figure 5-12.

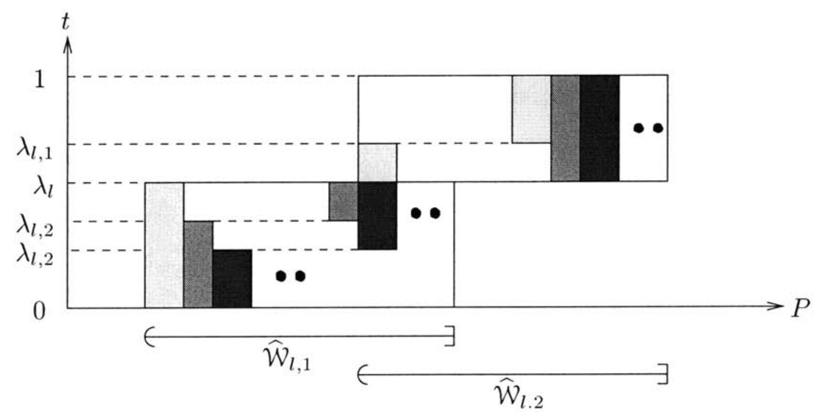


Figure 5-13: The construction outcome of the second set of operations in the second iteration using a different construction

[6]Upon closer examinations, it seems that this construction approach is identical to Rimoldi's result on time-sharing. In his work, it was shown that this construction is possible. In comparison, this work gives the details of the segmentation, and such details are made possible because of the visualization provided by the power diagram based frameworks.

Chapter 6

Conclusions

In this thesis, we approached the AWGN multi-access channel from a new perspective. We demonstrated that this perspective provides a unified framework for understanding this channel by studying both achievability and decoding issues. The outcome of this study is a series of simplifying results that lead to the conclusion that the N-user AWGN multi-access channel is no more complex than a set of $O(N)$ AWGN single-user channels with an ordering.

We began the introduction of this new perspective with the achievability studies in Chapter 2. At the foundation of this new perspective is a different way of associating the specifying parameters in the system with the channel and users. Comparing to the conventional way of associating these parameters, the new way reduces the scope of the channel so that it is specified by a single parameter – channel noise variance.

One consequence of the new association is that introducing additional users into the system no longer affects the channel specifications. In fact, the channel for two AWGN multi-access channels, one with N_1 users and the other with N_2 users, are identical if they have identical noise variances.

In comparison, such an introduction causes the dimensionality of the channel specification to increase under the conventional associations. This dimensionality expansion significantly complicates understanding the AWGN multi-access channel.

Moreover, the relative independence between the channel and the users achieved by the new association makes it possible to formulate and study properties of the set of users independently of the actual channel. As a result, many questions of concern can be reformulated as the comparison between channel specifications and user properties. For example,

1. the AWGN multi-access achievability is reformulated in Chapter 2 as the comparison of the channel noise variance with the *maximal noise threshold* for the given set of user specifications to be achievable

2. the successive decodability of the given set of user in the channel is reformulated in Chapter 3 as the comparison of the channel noise variance with the *maximal noise threshold* for the given set of user specifications to be SSD

3. the successive decodability of the given set of user in the channel with additional techniques (such as power-reduction, user-splitting, and time-sharing) is reformulated in Chapters 3, 4, and 5 as the comparison of the channel noise variance with the *maximal noise threshold* for the given set of user specifications to be successively decodable with the additional techniques

Such re-formulations achieve some immediate conceptual simplifications over their conventional formulations because the channel specification is excluded from the computation of the relevant property of the set of users.

Alternatively, the above approach can be viewed, under the new associations, as first selecting the one channel (rather, one channel noise variance) which barely allows the set of user specifications to be achievable, successively decodable, or successively decodable with some additional techniques; and then comparing the noise variance of the actual channel with that of the selected channel[1].

Having re-formulated these questions as a comparison between the channel noise variance and the maximal noise threshold of a given set of users, we note that the computational burden for solving these problems resides in computing the maximal noise thresholdWe considered simplifying structures (or properties) within a set of user specifications for such computations.

As seen in Chapter 2, the convexity and monotonicity in the relationship between the set of user specifications (power and rate) for a single user and its relevant maximal noise threshold, i.e. the channel capacity formula, make it possible to represent the two specifications for each user uniquely as an interval, each on the same axis. We call the former a *power interval*, and the latter, the *power axis* (or equivalently, a rate interval and the rate axis, which we saw in Chapter 3). Such a representation of multiple power intervals on a single power axis, which we call the *power diagram*, provides a convenient way to abstract and visualize the desired simplifying structures. We found that,

1. for AWGN multi-access achievability, the simplifying structure is the *overlapping* property;

2. for successive decodability with Gaussian ensemble of codes (SSD), the simplifying structure is the p-*overlapping* property;

[1]Built into this argument is the understanding that reducing the noise variance in a channel can only make decoding easier.

3. for power-reduced successive decodability with Gaussian ensemble of codes (PR-SD), the simplifying structure is the dual of the p-overlapping property in the rate diagram, called the r-*overlapping* property; and

4. for successive decodability with both user-splitting and time-sharing, the simplifying structure is again the *overlapping* property.

Using the power diagram framework, we were able to show that, in each of the studies, a set of user specifications with the appropriate simplifying property is equivalent to a single user for the purpose of the study. In other words, if a given set of users is asserted to possess the desired simplifying property, then the desired maximal noise threshold property of the set can be obtained in the same manner as that for a single user, which was shown, from the respective chapters, to be a single computation.

Our study led to algorithms, all based on the power diagram framework, that checked each of these simplifying properties for a given N-user set requiring no more than $O(N \ln N)$ computations. Recall that,

1. Algorithm 2.3.8 is used for checking the overlapping property;

2. Algorithm 3.2.5 is used for checking the p-overlapping property; and

3. Algorithm 3.3.7 is used for checking the r-overlapping property.

In fact, the computational burden of all of the algorithms are dominated by an initial sort; and the remaining computations in all cases amounts to $O(N-1)$. These algorithms led to the conclusion that each of the questions posed above can be solved, if the given set of users possess the desired simplifying property, with a set of $O(N \ln N)$ computations.

Now, in general, a given set of user specifications may or may not possess the desired simplifying property. We considered partitioning such a set of user specifications into blocks of greatest (in terms of number of users) subsets with the appropriate simplifying property. In particular,

1. for computing the maximal noise threshold for achievability and successive decodability with either user-splitting or time-sharing, the partition scheme is associated with the *irreducible equivalent* construction;

2. for computing the maximal noise threshold for SSD, the partition scheme is associated with the p-*maximal allocation set* construction; and

3. for computing the maximal noise threshold for PR-SD, the partition scheme is associated with the r-*maximal allocation set* construction.

Computing the desired maximal noise threshold property after such a partitioning was shown to be no more demanding than computing such a property for a single user.

It turns out that, when the algorithms used for checking the desired simplifying property determines that a given set of user specifications does not possess the property, it provides the desired partitioning[2]. Hence, such general cases do not require more computations. In fact, computing the irreducible equivalent for a non-overlapping set actually requires fewer than $O(N-1)$ computations after the initial sort.

Collectively, these results demonstrate that, from the achievability point-of-view, the N-user AWGN multi-access channel is no more complex than $O(N)$ independent single-user channels with an ordering. Compared to the brute force method of checking the N-user AWGN multi-access achievability by $O(2^N)$ computations, these results represent dramatic simplifications.

This new perspective, particularly the power diagram framework, was also demonstrated as a simplifying approach to decoding in AWGN multi-access channels. In particular, we studied *simple successive decoding*, *power-reduced successive decoding*, *successive decoding with user-splitting*, and *successive decoding with time-sharing*.

One advantage of studying decoding techniques using the power diagram is that the latter presents a resource-allocation approach. Such an approach makes it possible to identify the optimal successive decoding order in both simple successive decoding and power-reduced successive decoding by a simple sort of the appropriate characteristics of the users (see Chapter 3. From this point-of-view, the simplification of achievability with the above techniques is a direct consequence of the simplicity in discovering the optimal decoding order. Additionally, we were able to derive a complete characterization of the set of rate tuples achievable using power-reduced successive decoding.

The other advantage of the power diagram is that it presents all relevant user specifications graphically on a one-dimensional axis. Hence, we are able to visualize the necessary manipulations directly and simply. This is particularly useful in studies of combination successive decoding techniques, i.e. successive decoding with either user-splitting or time-sharing. For the former (successive decoding with user-splitting), we were not only able to extend the existing user-splitting constructions to arbitrary sets of achievable user specifications in a natural way, but also were able to create a set of results that encompasses all possible user-splitting constructions for successive decoding. For the latter (successive decoding with time-sharing), we extended the power diagram into the time-power diagram by representing the time-sharing

[2]Another way to see this is that when a given set of users do possess the desired property, the algorithm would declare that there exists a single partition – the entire set.

characteristics on a time-axis. As the result, we were able to show that no more than $2N-1$ successive decoding steps are required.

Since both combination successive decoding techniques can be used to decode arbitrary sets of achievable user specifications, these results led to the conclusion that decoding an N-user AWGN multi-access channel may not require more computations than decoding $O(N)$ independent AWGN single-user channels.

At the completion of this thesis, we have firmly established that, conceptually, an N-user AWGN multi-access channels is no more complex than $O(N)$ independent AWGN single-user channels in its achievability and decoding. However, as mentioned in the introduction chapter, such simplifications, particularly in the decoding area, are not readily adaptable to practice. This is because our analysis follows the theoretical assumption that the transmission of each user may be decoded to arbitrarily small probability of error, which permits the next successive decoding step to be accomplished without any interference from the user just decoded. In reality, this is almost never the case because infinite delay is not permissible. Hence the errors from previously decoded transmissions may cause additional errors in subsequent decoding steps (a phenomenon called error propagation). The error probability due to this error propagation can be easily upper bounded, but we hope to achieve improved bounds on this error probability in the not-too-distant future.

Appendix A

An Alternate Proof of Theorem 2.2.1

Theorem 2.2.1. *Let T_1, T_2 be two power intervals, and T be their combined user, i.e. $T = \mathsf{T}(\{T_1, T_2\})$. If $T_1 \cap T_2 \neq \emptyset$, then $\overline{\mathsf{ext}(\{T_1, T_2\})} \subset T^o$. If T_1 and T_2 are adjacent, then their combined user is equal to their extent and their union, i.e. $T = T_1 \cup T_2$. Otherwise, T_1 and T_2 are disjoint, and $\overline{T} \subset \mathsf{ext}(\{T_1, T_2\})^o$.*

Proof. Let $T_i = (a_i, b_i], i = 1..2$ and $T = \mathsf{T}(\{T_1, T_2\}) = (a, b]$.

By definition of combined user, there are

$$\mathsf{p}(T) = \mathsf{p}(T_1) + \mathsf{p}(T_2) \tag{A.1}$$

$$\mathsf{r}(T) = \mathsf{r}(T_1) + \mathsf{r}(T_2) \tag{A.2}$$

In other words, there are

$$b - a = b_1 - a_1 + b_2 - a_2 \tag{A.3}$$

$$\frac{1}{2}\ln\left(\frac{b}{a}\right) = \frac{1}{2}\ln\left(\frac{b_1}{a_1}\right) + \frac{1}{2}\ln\left(\frac{b_2}{a_2}\right) \tag{A.4}$$

Multiply both sides of (A.4) by 2 and take the anti-log:

$$\frac{b}{a} = \frac{b_1 b_2}{a_1 a_2} \tag{A.5}$$

Since $b = a + b_1 - a_1 + b_2 - a_2$ from the above, we have:

$$1 + \frac{b_1 - a_1 + b_2 - a_2}{a} = \frac{b_1 b_2}{a_1 a_2} \tag{A.6}$$

Subtract 1 from both sides, and solve for a, we have:

$$a = \frac{b_1 - a_1 + b_2 - a_2}{\frac{b_1 b_2}{a_1 a_2} - 1} \tag{A.7}$$

Hence,

$$\begin{aligned} a - a_1 &= \frac{b_1 - a_1 + b_2 - a_2}{\frac{b_1 b_2}{a_1 a_2} - 1} - a_1 \\ &= \frac{(b_1 - a_1 + b_2 - a_2)\, a_2 - (b_1 b_2 - a_1 a_2)}{\frac{b_1 b_2}{a_1} - a_2} \\ &= \frac{(b_2 - a_2)\, a_2 - b_1 (b_2 - a_2)}{\frac{b_1 b_2}{a_1} - a_2} \\ &= \frac{(b_2 - a_2)(a_2 - b_1)}{\frac{b_1 b_2}{a_1} - a_2} \\ &= \frac{a_1 (b_2 - a_2)(a_2 - b_1)}{b_1 b_2 - a_1 a_2} \end{aligned} \tag{A.8}$$

Since $b_2 > a_2$,

$$a < a_1 \iff a_2 < b_1 \tag{A.9}$$

Similarly,

$$\begin{aligned} b - b_1 &= a + b_1 - a_1 + b_2 - a_2 - (a_1 + b_1 - a_1) \\ &= a - a_1 + b_2 - a_2 \\ &= \frac{(a_2 - b_1)(b_2 - a_2)}{\frac{b_1 b_2}{a_1} - a_2} + b_2 - a_2 \\ &= \frac{(b_2 - a_2)}{\frac{b_1 b_2}{a_1} - a_2} \left((a_2 - b_1) + \frac{b_1 b_2}{a_1} - a_2 \right) \\ &= \frac{(b_2 - a_2)}{b_1 b_2 - a_1 a_2} \left(a_1 (a_2 - b_1) + b_1 b_2 - a_1 a_2 \right) \\ &= \frac{b_1 (b_2 - a_2)(b_2 - a_1)}{b_1 b_2 - a_1 a_2} \end{aligned} \tag{A.10}$$

since $b_2 > a_2$,

$$b > b_1 \iff a_1 < b_2 \tag{A.11}$$

Similarly,

$$a - a_2 = \frac{a_1 (b_1 - a_1)(a_1 - b_2)}{b_1 b_2 - a_1 a_2} \tag{A.12}$$

$$b - b_2 = \frac{b_2 (b_1 - a_1)(b_1 - a_2)}{b_1 b_2 - a_1 a_2} \tag{A.13}$$

Since $b_1 > a_1$,

$$a < a_2 \iff a_1 < b_2 \tag{A.14}$$

$$b > b_2 \iff a_2 < b_1 \tag{A.15}$$

Combine (A.9) and (A.14),

$$a < \min\{a_1, a_2\} \iff a_1, a_2 < b_1, b_2 \tag{A.16}$$

Similarly combining (A.9) and (A.14),

$$b > \max\{b_1, b_2\} \iff a_1, a_2 < b_1, b_2 \tag{A.17}$$

Since the condition is if and only if, this proves the theorem.

□

Appendix B

An Alternate Introduction to the Overlapping Property

Let n_o be the noise intensity in the given AWGN multi-access channel and $\mathcal{U} = \{T_i, i \in [1, \dots, N]\}$ be the set of N-user specifications. The achievability of $\mathcal{U}$ in this channel is conventionally checked by the following set of conditions:

$$\sum_{T_i \in \mathcal{V}} \mathsf{r}(T_i) \leq \frac{1}{2} \ln \left(\frac{n_o + \sum_{T_i \in \mathcal{V}} \mathsf{p}(T_i)}{n_o} \right), \quad \forall \mathcal{V} \subseteq \mathcal{U} \tag{B.1}$$

In the power diagram framework, for each $\mathcal{V} \subseteq \mathcal{U}$, we may regard $\sum_{T_i \in \mathcal{V}} \mathsf{p}(T_i)$ and $\sum_{T_i \in \mathcal{V}} \mathsf{r}(T_i)$ as the power and rate specification of the combined user $\mathsf{T}(\mathcal{V})$, and transform the condition to

$$n_o \leq \mathsf{T}(\mathcal{V}) \tag{B.2}$$

Hence the *maximal tolerable channel noise intensity of $\mathcal{U}$* is the minimum of the lower boundary of $\mathsf{T}(\mathcal{V})$ for all $\mathcal{V} \subseteq \mathcal{U}$.

Recall that the maximal tolerable channel noise intensity of a two-user set depends on whether the power intervals of the two users are *either intersecting or adjacent*. Specifically, if the two power intervals are intersecting or adjacent, then the maximal tolerable channel noise intensity is the lower boundary of their combined user. Otherwise, the maximal tolerable channel noise intensity is the minimum of their lower boundaries.

In this section, we establish an *overlapping* property (to be defined later) that plays a similar role in simplifying the N-user achievability conditions as the *intersection or adjacent* property does in the two-user case. Not surprisingly, this overlapping property is closely related to the intersection property. As will be seen shortly, the algorithm that checks whether a set of user specifications has the overlapping property iteratively operates on two intersecting or adjacent power intervals.

In the following, we first introduce this algorithm. We then identify the overlapping property and prove its consequences on simplifying the N-user achievability conditions. Finally, we detail and analyze the resulting algorithms that check the N-user achievability with $O(N\ln(N))$ computations.

For the given set of power intervals $\mathcal{U}$, the following algorithm specifies an order to successively replace two power intervals that are either adjacent or intersecting by their combined power interval.

Algorithm B.0.7.

Step 1: *Order members of* $\mathcal{U}$ *into* $\mathcal{U}' = \{T_i = (a_i, b_i], i \in [1, \ldots, N]\}$ *such that* $a_i \leq a_{i+1}$*; initialize* $i = 1$.

Step 2: *If* T_i *is the last element in* $\mathcal{U}'$*, exit the algorithm. Otherwise*

Step 3: *If* T_i *and* T_{i+1} *are disjoint, i.e.* $b_i < a_{i+1}$*, then increase* i *by* 1 *and go to Step 2. Otherwise*

Step 4: T_i *and* T_{i+1} *are either adjacent or intersecting. Replace the two power intervals by their combined user* $\mathsf{T}(\{T_i, T_{i+1}\})$*, call the new power interval* T_i *and update* $\mathcal{U}'$ *by renaming* $T_j = T_{j+1}$ *for all* $j \geq i+1$*. If* T_{i-1} *exists, decrease* i *by* 1*, and go to Step 3; else go to Step 2*

Figure B-1 presents a flowchart of one iteration of this algorithm.

As seen in Figure B-1, Step 4 of the algorithm can be seen as creating a parent node for the two (adjacent or intersecting) power intervals that is equal to their combined users. Thus successive iteration of this step in the algorithm can be visualized as the inverted growth of a forest of binary trees. In the special case when the outcome of this algorithm is a singleton set, a single binary tree is constructed.

With this visualization of the algorithm in mind, we make the following observations:

Observation 1: Once two adjacently ordered power intervals are combined (in Step 4), they both cease to participate in further constructions, because the algorithm proceeds with the newly created parent node. Hence, no node with parents participates in further constructions.

Observation 2: By Theorem 2.2.1, the combined user created in Step 4 contains the two power intervals. Repeating this argument, we see that any intermediate node on the

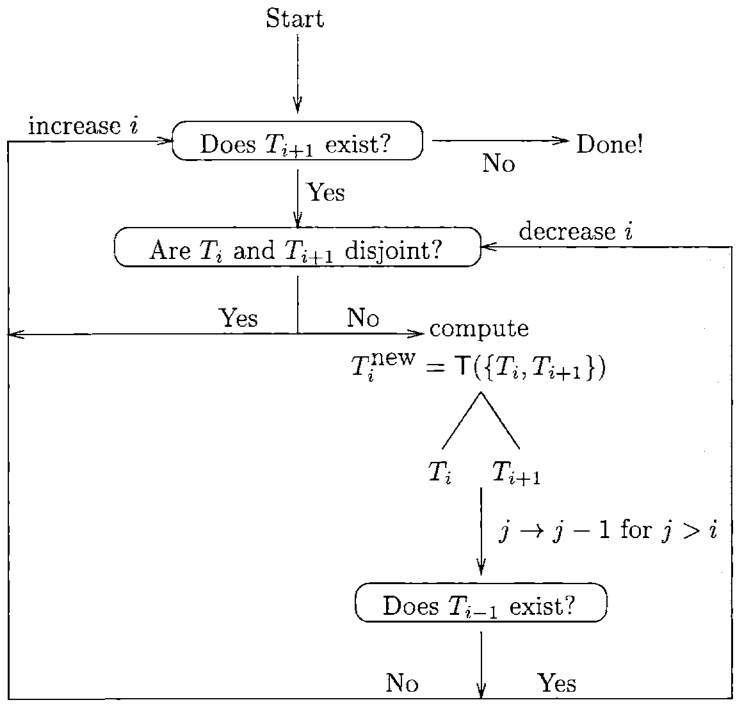


Figure B-1: Flowchart of One Iteration of Algorithm B.0.7 (boldface indicates the power interval of interest for the next iteration)

tree must contain all of its descendent power intervals. Moreover, the resulting set at every iteration of the algorithm must have the same combined user as the original set. In particular, the final outcome of this algorithm has the same combined user as the original set.

Observation 3: In Step 4, after the combined user is created and replaces the two power intervals, i is decremented before going to Step 2. This takes care of the case when the combined user power interval is non-disjoint with the previous power interval. Therefore, the outcome of this algorithm must be either a singleton set or a set of disjoint power intervals.

Observation 4: Moreover, since only the previous power interval is visited after the combining step, the member power intervals in the outcome set must be created in order from the lowest to the highest such that no combining step may take place for the next (higher) member of the outcome set until the construction of the current one finishes.

The outcome of the algorithm is clearly unique. We define the following.

Definition B.0.7. The resulting set of Algorithm B.0.7, when its members are ordered in the ascending order according to Definition 2.1.3, is called the *irreducible equivalent* of $\mathcal{U}$, denoted by $\mathsf{irr}(\mathcal{U})$.

The i-th member power interval in $\mathsf{irr}(\mathcal{U})$ is denoted by $[\mathsf{irr}\,(\mathcal{U})]_i$.

The set of leaf power intervals that are descendents of $[\mathsf{irr}\,(\mathcal{U})]_i$ is termed the *i-th constructing set* of $\mathsf{irr}(\mathcal{U})$, and denoted by $\mathsf{B}_i(\mathcal{U})$.

By definition, $[\mathsf{irr}\,(\mathcal{U})]_i$ is the combined user of $\mathsf{B}_i(\mathcal{U})$.

With this definition, Observation 4 states that members of the irreducible equivalent are constructed in order starting from $[\mathsf{irr}\,(\mathcal{U})]_1$.

Similarly, we call the forest of binary trees constructed by Algorithm B.0.7 the *irreducible equivalent forest*, and the individual trees the *irreducible equivalent tree*. Hence, a binary tree is an irreducible equivalent tree if the following two conditions are satisfied:

Condition 1: The two immediate children of every parent node have non-disjoint power intervals, and every parent node corresponds to a power interval that is the combined user of its two immediate children.

Condition 2: The order of combining (the parentage) follows that of Algorithm B.0.7.

Now, we define the overlapping property:

Definition B.0.8. If $\mathsf{irr}(\mathcal{U})$ is a singleton set, we say that $\mathcal{U}$ has the *overlapping property*, or simply that $\mathcal{U}$ is an *overlapping set*.

In other words, an overlapping set corresponds to the special case when Algorithm B.0.7 results in a singleton set. Therefore, its member are the leaf nodes of a single irreducible equivalent tree.

By definition, a singleton set has the overlapping property. Moreover, Theorem 2.2.1 provides that two power intervals that are either intersecting or adjacent form an overlapping set. Hence, the two power intervals that are combined in Step 4 of Algorithm B.0.7 form an overlapping set. Figure B-2 illustrates the two possible irreducible equivalent trees for an overlapping set of 3 power intervals.

We formalize Observations 2 and 4 into the following lemma and theorem.

Lemma B.0.8. *The combined power interval of an overlapping set contains each member of the set.*

Proof. Let $\mathcal{U}$ have the overlapping property. By definition, members of $\mathcal{U}$ must be the leaf nodes of an irreducible equivalent tree. Since a parent node of an irreducible equivalent tree is the combined user of its two immediate (overlapping) children, Theorem 2.2.1 shows that a

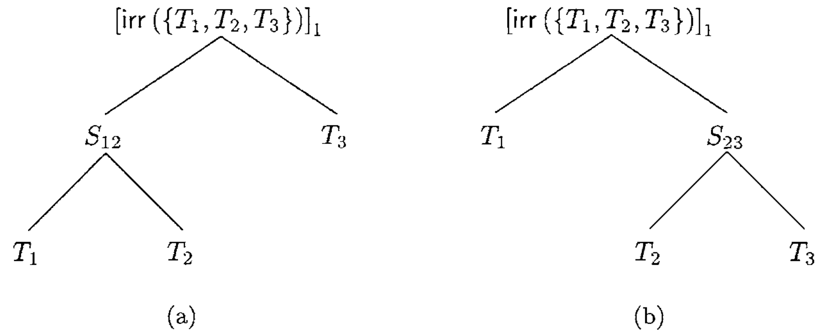


Figure B-2: The Two Possible Irreducible Equivalent Trees of 3 Overlapping Power Intervals

parent node power interval must contain its immediate children. Repeating this argument, the proof is complete. □

Theorem B.0.9. *Every constructing set has the overlapping property.*

Proof. Since Algorithm B.0.7 proceeds from the member of $\mathcal{U}$ with the least lower boundary, $\mathsf{B}_1(\mathcal{U})$ clearly has the overlapping property.

Observe that the power interval in $\mathsf{B}_2(\mathcal{U})$ with the least lower boundary is reached only after $[\mathsf{irr}\,(\mathcal{U})]_1$ is constructed. Moreover, once $[\mathsf{irr}\,(\mathcal{U})]_1$ is constructed, it remains disjoint from any power intervals constructed during further operations of Algorithm B.0.7. In other words, the first member of $\mathsf{irr}(\mathcal{U} \setminus \mathsf{B}_1(\mathcal{U}))$ is $[\mathsf{irr}\,(\mathcal{U})]_2$. Hence $[\mathsf{irr}\,(\mathcal{U})]_2$ also has the overlapping property.

Repeating this argument, we complete the proof of this theorem. □

In other words, the constructing sets of the irreducible equivalent partition a set of power intervals into overlapping subsets.

The overlapping property is the key to simplifying the N-user achievability conditions. Consider the following consequence of the overlapping property.

Theorem B.0.10 (The Containment Theorem). $\mathsf{T}(\mathcal{V}) \subseteq \mathsf{T}(\mathcal{U})$ *for all* $\mathcal{V} \subseteq \mathcal{U}$ *if and only if* $\mathcal{U}$ *has the overlapping property.*

The proof of this theorem hinges on the following property of the overlapping sets.

Theorem B.0.11. *Let* $\mathcal{U}$ *be an overlapping set, and* $\mathcal{V}$ *a subset of* $\mathcal{U}$ *that also has the overlapping property. Then* $\mathcal{U} \setminus \mathcal{V} \cup \{\mathsf{T}(\mathcal{V})\}$ *has the overlapping property.*

Remark. The proof of this theorem uses a graphical construct that provides much insight into the structural properties of sets of power intervals, not just for simplifying achievability.

Proof. The proof is aided by the irreducible equivalent tree visualization.

Since $\mathcal{U}$ is an overlapping set, by definition, there exists an irreducible equivalent tree with all members of $\mathcal{U}$ as the leaves. For brevity, we will refer to this tree as the *$\mathcal{U}$-tree*.

Since $\mathcal{V}$ is an overlapping set, by definition, there exists an irreducible equivalent tree with all members of $\mathcal{V}$ as the leaves. For brevity, we will refer to this tree as the *$\mathcal{V}$-tree*.

One way to show that $\mathcal{U} \setminus \mathcal{V} \cup \{\mathsf{T}(\mathcal{V})\}$ has the overlapping property is to construct an irreducible equivalent tree with all members of this set as the leaves. Recall the two conditions for a binary tree to be an irreducible equivalent tree, we accomplish this task in two steps: first, we *morph* the *$\mathcal{U}$-tree* into a new rooted binary tree which contains the *$\mathcal{V}$-tree* as a subtree that satisfies Condition 1 of irreducible equivalent tree, i.e. every parent node has two overlapping immediate children and is the combined user of the two. Then, we replace this subtree by its root node, and proceed to *morph* the result to satisfy Condition 2 of irreducible equivalent tree, i.e. to create the appropriate parentage.

It turns out that both steps are based on successively applying the same *morphing procedure*. This procedure morphs a given binary tree into a new one on which two arbitrarily chosen overlapping leaf power intervals share the same immediate parent, while preserving Condition 1.

On a binary tree satisfying Condition 1, let T_1 and T_2 be two overlapping leaf power intervals descending from different parents. This procedure starts with the following identification as illustrated in Figure B-3:

- identify the immediate parent node of T_1 as node C;
- identify the node that is the first common parent node of T_1 and T_2 as node D; and
- identify the parents of T_1 and T_2 which are the immediate children of D as nodes A and B, respectively.

The actual procedure takes 4 steps.

1. sever the branch connecting nodes B and D; and sever the branch connecting T_1 to node C;
2. replace the node T_2 with a new node and make T_1 and T_2 children of the new node;
3. on the original tree, collapse nodes A and D by replacing node D with node A;
4. re-attach the severed branch as the missing children of node C; and recompute power intervals at the appropriate node of the resulting tree.

The new binary tree satisfies Condition 1 because of the following.

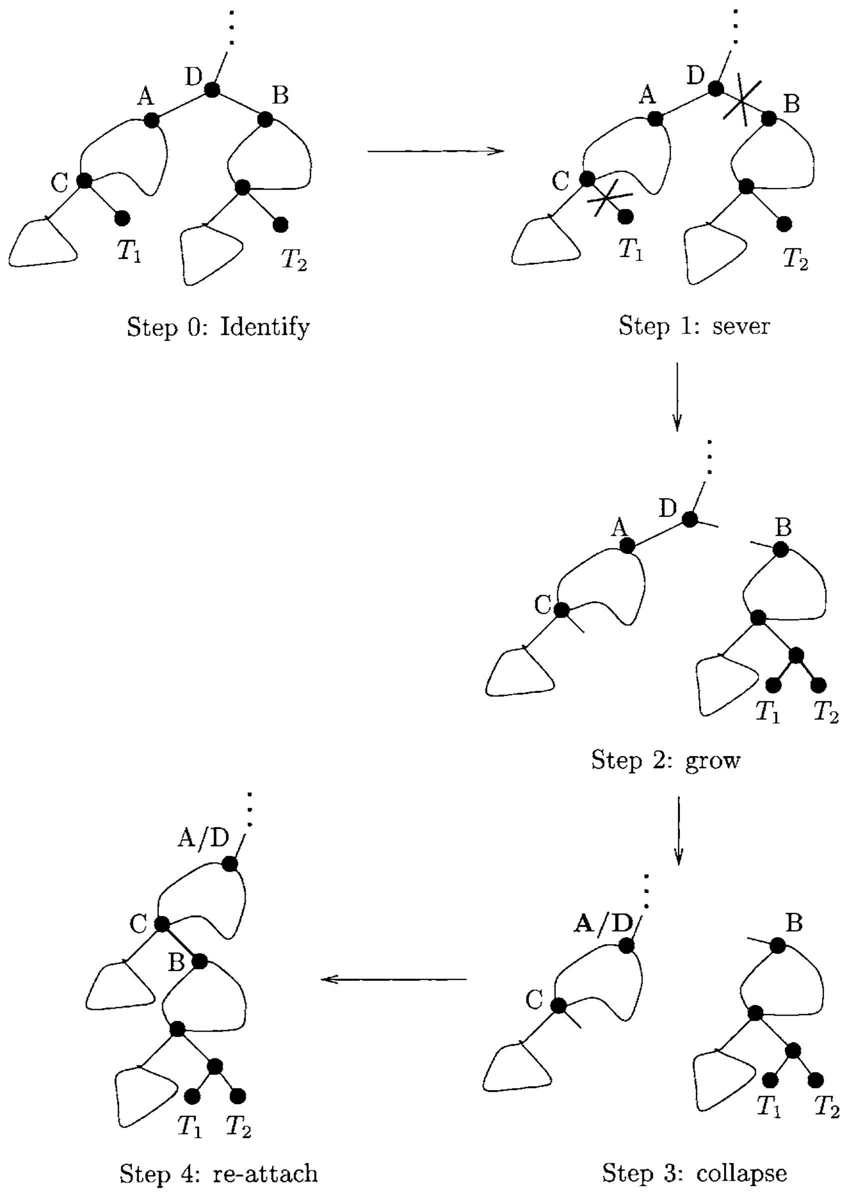


Figure B-3: The *Morphing* Procedure

- The original leaf node T_2 is replaced by the combined power interval of T_1 and T_2. By Theorem 2.2.1, $\mathsf{T}(\{T_1, T_2\})$ contains T_2. Repeatedly applying Theorem 2.2.1, we see that the two immediate children at every node on the new tree descending from B must overlap. Hence, the new power interval at node B must also contain both T_1 and T_2.

- Since the new power interval at node B contains T_1, the two immediate children at every node on the new tree descending from node A/D must also overlap.

- Since any descendent of node D on the original tree remains a child leaf of node A/D, the power interval at node A/D on the new tree is identical to the power interval at node D on the original tree, hence the immediate children on every node on the rest of tree (from node A/D up) overlap.

Now, to accomplish the first step, we observe that every leaf on the $\mathcal{V}$*-tree* is a leaf on the $\mathcal{U}$*-tree*. Hence we only successively apply the *morphing procedure* to the $\mathcal{U}$*-tree* following the $\mathcal{V}$*-tree* from leaves up.

By Observation 1, once the parent is created, neither children participates in further constructions. Hence, the outcome of replacing the $\mathcal{V}$*-subtree* by its root node continues to satisfy Condition 1.

To complete the proof, we need to *morph* this tree into one that has the same parentage as that resulting from Algorithm B.0.7. We observe that every new node created by Algorithm B.0.7 must be the immediate parent of two overlapping nodes. Hence the *morphing* procedure discussed in the above can be applied for such creation. Repeating this argument successively, we complete this proof. □

Repeatedly applying this theorem, we have the following corollary.

Corollary B.0.12. *Given an overlapping set of users* $\mathcal{U}$*, let* $\mathcal{V}_i$*, for* $i \in [1, \dots, K]$*, be disjoint subsets of* $\mathcal{U}$*, each having the overlapping property. Then*

$$\mathcal{U} \setminus \left(\cup_{i=1}^{K} \mathcal{V}_i\right) \cup \{\mathsf{T}(\mathcal{V}_i), i \in [1, \dots, K]\} \tag{B.3}$$

has the overlapping property.

Since constructing sets have the overlapping property, we also have the following.

Corollary B.0.13. *Let* $\mathcal{V}_i$*, for* $i \in [1, \dots, K]$ *be disjoint subsets of a set of power intervals* $\mathcal{U}$*, each having the overlapping property. Then*

$$\mathsf{irr}\left(\mathcal{U} \setminus \left(\cup_{i=1}^{K} \mathcal{V}_i\right) \cup \{\mathsf{T}(\mathcal{V}_i), i \in [1, \dots, K]\}\right) = \mathsf{irr}(\mathcal{U}) \tag{B.4}$$

Theorem B.0.11 and its corollaries effectively state that it is not necessary to follow the order of creating parent nodes in Algorithm B.0.7. In fact, an overlapping set is a set of power intervals whose members are the leaf nodes of any binary tree that satisfies Condition 1. In other words, the following algorithm has identical results as Algorithm B.0.7.

Algorithm B.0.14.
Let $\mathcal{U}' = \mathcal{U}$.

1. *if there exist* T_1, $T_2 \in \mathcal{U}'$ *such that* T_1 *and* T_2 *are non-disjoint (i.e. either intersecting or adjacent), then let* $\mathcal{U}' = \mathcal{U}' \setminus \{T_1, T_2\} \cup \{\mathsf{T}(\{T_1, T_2\})\}$, *and repeat this step*

2. *otherwise, stop*

Observe that one alternate realization of Algorithm B.0.7 is to construct the irreducible equivalent from the member power interval with the greatest upper boundary down. (We will see more construction with the upper ends of the power intervals in the next chapter.)

For simplicity, we shall refer to any binary tree that satisfy Condition 1 as an *irreducible equivalent tree*; and a multitude of such trees as an *irreducible equivalent forest.*

Incidentally, the above results also gives that $\mathsf{B}_i(\mathcal{U})$ is the set of all members of $\mathcal{U}$ that are contained in $[\mathsf{irr}(\mathcal{U})]_i$. Specifically,

$$\mathsf{B}_i(\mathcal{U}) = \{T \in \mathcal{U} : T \subseteq [\mathsf{irr}(\mathcal{U})]_i\} \tag{B.5}$$

We now prove the containment theorem.

Proof of Theorem B.0.10: Suppose $\mathsf{irr}(\mathcal{V})$ contains M elements. By definition of the irreducible equivalent,

$$\mathcal{V} = \bigcup_{j=1}^{M} \mathsf{B}_j(\mathcal{V}) \tag{B.6}$$

By Theorem B.0.9, $\mathsf{B}_j(\mathcal{V})$ is an overlapping set for each $j \in [1, \ldots, M]$. Hence we may successively apply Theorem B.0.11 for each j to conclude that

$$\mathcal{U} \setminus \left(\cup_{j=1}^{M} \mathsf{B}_j(\mathcal{V})\right) \cup \left(\cup_{j=1}^{M} \{[\mathsf{irr}\,(\mathcal{V})]_j\}\right) = \mathcal{U} \setminus \mathcal{V} \cup \mathsf{irr}(\mathcal{V}) \tag{B.7}$$

is an overlapping set. Hence by Lemma B.0.8, $[\mathsf{irr}(\mathcal{V})]_j \subseteq \mathsf{T}(\mathcal{U})$ for all j. Therefore,

$$\mathsf{ext}(\mathsf{irr}(\mathcal{V})) \subseteq \mathsf{T}(\mathcal{U}) \tag{B.8}$$

If $\mathsf{irr}(\mathcal{V})$ is a singleton set, there is nothing further to prove.

Otherwise, $\mathsf{irr}(\mathcal{V})$ is a set of disjoint power intervals. By Corollary 2.2.4, we have

$$\overline{\mathsf{T}(\mathsf{irr}(\mathcal{V}))} \subset \mathsf{ext}(\mathsf{irr}(\mathcal{V}))^o \tag{B.9}$$

Combining with (B.8), we have

$$\mathsf{T}\left(\mathsf{irr}(\mathcal{V})\right) \subset \mathsf{T}(\mathcal{U}) \tag{B.10}$$

Finally, we notice that $\mathsf{T}(\mathsf{irr}(\mathcal{V})) = \mathsf{T}(\mathcal{V})$ by definition, and completes the forward part of the proof.

For the converse, suppose $\mathcal{U}$ does not have the overlapping property. Then $\mathsf{irr}(\mathcal{U})$ is a set of disjoint power intervals. By Corollary 2.2.4, we have

$$\overline{\mathsf{T}(\mathsf{irr}(\mathcal{U}))} \subset \mathsf{ext}(\mathsf{irr}(\mathcal{U}))^o \tag{B.11}$$

Since $\mathsf{T}(\mathcal{U}) = \mathsf{T}(\mathsf{irr}(\mathcal{U}))$ by definition, we have

$$\overline{\mathsf{T}(\mathcal{U})} \subset \mathsf{ext}(\mathsf{irr}(\mathcal{U}))^o \tag{B.12}$$

In particular, the lower boundary of $[\mathsf{irr}(\mathcal{U})]_1$ must be below the lower boundary of $\mathsf{T}(\mathcal{U})$ (and the upper boundary of the highest element of $\mathsf{irr}(\mathcal{U})$ must be higher than the upper boundary of $\mathsf{T}(\mathcal{U})$). This proves the converse. □

One interpretation of the containment theorem is that:

Corollary B.0.15. *The* maximal tolerable channel noise intensity *of an overlapping set is the lower boundary of the set's combined power interval.*

Corollary B.0.16. *The maximal tolerable noise intensity of $\mathcal{U}$ is the lower boundary of* $[\mathsf{irr}(\mathcal{U})]_1$.

Proof. Theorem B.0.10 proves the case when $\mathsf{irr}(\mathcal{U})$ is a singleton set.

Let the lower boundary of $[\mathsf{irr}(\mathcal{U})]_1$ be a. Recall that $\{\mathsf{B}_i(\mathcal{U}), i \in [1, \dots, M]\}$, where $M > 1$ is the number of power intervals in $\mathsf{irr}(\mathcal{U})$, is a partition of $\mathcal{U}$.

Let $\mathcal{V}$ be a subset of $\mathcal{U}$. Construct $\mathcal{V}_i = \mathcal{V} \cap \mathsf{B}_i(\mathcal{U})$ for each i. Then $\{\mathcal{V}_i, i \in [1, \dots, M]\}$ is a partition of $\mathcal{V}$. Theorem B.0.10 provides that for each $i \in [1, \dots, M]$, we have

$$\mathsf{T}(\mathcal{V}_i) \subseteq [\mathsf{irr}(\mathcal{U})]_i \tag{B.13}$$

Since $\mathsf{irr}(\mathcal{U})$ is disjoint (as $M > 1$), $\{\mathsf{T}(\mathcal{V}_i), i \in [1, \dots, M]\}$ must also be disjoint.

By Corollary 2.2.4, we have

$$\overline{\mathsf{T}(\{\mathsf{T}(\mathcal{V}_i), i \in [1, \dots, M]\})} \subset \mathsf{ext}(\{\mathsf{T}(\mathcal{V}_i), i \in [1, \dots, M]\})^o \subseteq \mathsf{ext}(\mathsf{irr}(\mathcal{U}))^o \tag{B.14}$$

where the last containment is by (B.13). Finally, since the lower boundary of $[\mathsf{irr}\,(\mathcal{U})]_1$ is a, we have this corollary. □

Accordingly, we have the following algorithm to determine the achievability of a finite set of user power intervals $\mathcal{U}$ in an AWGN multi-access channel with channel noise intensity n_o.

Algorithm B.0.17.

1. *Construct the irreducible equivalent of the given set of user power intervals; let the lower boundary of* $[\mathsf{irr}(\mathcal{U})]_1$ *be* η
2. *Compare* η *with* n_o*, and conclude that the given set of user specifications is achievable in this channel if and only if* $n_o \leq \eta$

The complexity of this algorithm depends on the complexity of the algorithm that constructs the irreducible equivalent. Suppose Algorithm B.0.7 is used. Observe that once the set of user power intervals are ordered by the lower end of the intervals, only one operation, comparing the upper end of the lower power interval with the lower end of the upper one, is performed (in Step 3) to determine whether the two adjacently ordered power intervals overlap. If they do overlap, the operation of replacing the two by their combined user power intervals reduces the number of users in the resulting set (for further construction) by 1. Therefore, no more than $N - 1$ such combination operations need to be performed for a set of N users. Since in each combining operation, there is one comparison with the next lower interval, there can be at most $2N - 3$ comparisons ($N - 1$ as the algorithm increment i, and $N - 2$ as the algorithm decrement i). Therefore, even if the user specifications are originally given in power constraints and transmission rates, the worst case computation complexity of the implementation using the above algorithm is dominated by the initial sort, which is $O(N \ln(N))$.

While it is possible to improve on the above algorithm to reduce the computational burden further, for example by observing that computing the combined user power interval for more than two users at once is simpler than successively combining two at a time, we conjecture that the worst case complexity can not be reduced below $O(N \ln(N))$.

Incidentally, this algorithm is also readily adaptable for evaluating incremental achievability, e.g. answering the question whether an additional user changes the system's achievability. Notice that exhaustively checking all conditions for N-user AWGN multi-access achievability

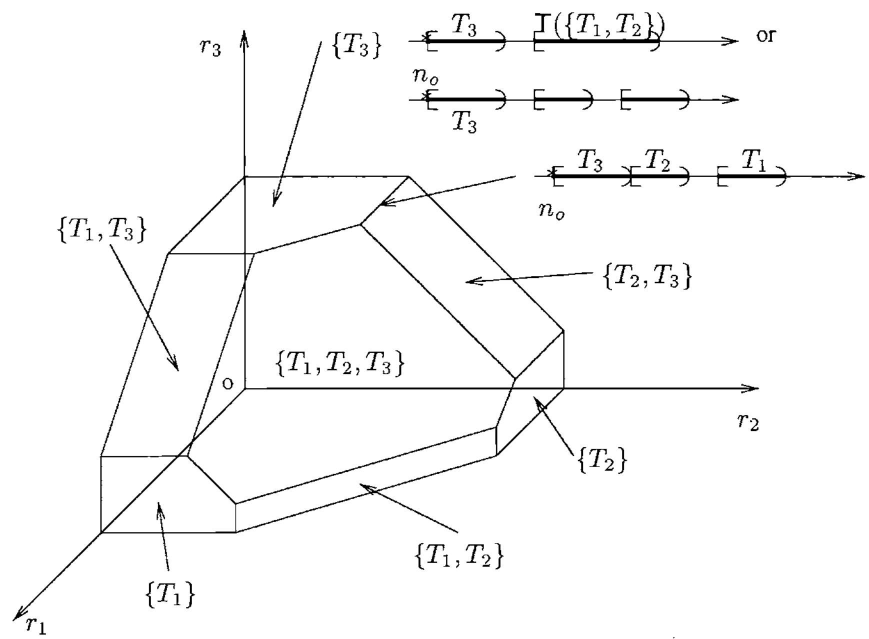


Figure B-4: The $\mathsf{B}_1(\mathcal{U})$ on the Bounding Surface of Three-User Achievable Rate Region

requires $2^N - 1$ checks for the additional user. In contrast, this method can compute the irreducible equivalent of the $N+1$-user set from that of the N-user set. Assuming the irreducible equivalent of the N users is already computed, the additional computation burden is bounded by $O(M)$, where M is the number of elements in the irreducible equivalent of the initial set. Even in the worst case when M is equal to N – when the initial set consists of disjoint power intervals – this method still achieves a substantial saving.

Insights developed in the previous section for checking the two-user achievability can be readily generalized to the N-user case. In particular, the irreducible equivalent of a set of N-user power intervals $\mathcal{U}$ determines whether the set corresponds to a rate tuple in the interior or on the boundary (on which bounding plane) of the achievable rate region of a given channel. Corollary B.0.16 states that the maximal tolerable channel noise intensity is the lower boundary of $[\mathsf{irr}\,(\mathcal{U})]_1$. Let the lower boundary of $[\mathsf{irr}\,(\mathcal{U})]_1$ be η. If the noise intensity of the given channel n_o is greater than η, then $\mathcal{U}$ is not achievable. If n_o is less than η, then $\mathcal{U}$ is achievable, and $\mathcal{U}$ corresponds to a rate tuple in the interior of the given channel's rate achievable region. Finally, if n_o is equal to η, then $\mathsf{B}_1(\mathcal{U})$ is an overlapping set with the lower boundary of its combined user at the channel noise intensity n_o. In this case, $\mathcal{U}$ corresponds to a rate tuple on the bounding plane of the given channel's rate achievable region where the combined rate of users in $\mathsf{B}_1(\mathcal{U})$ is the maximal. Figure B-4 illustrates this last case for a 3-user set.

Extending the last understanding, we see that for each $k \in [1, \ldots, M]$, the set of user specifications $\cup_{j \geq k} \mathsf{B}_j(\mathcal{U})$ corresponds to a point on the boundary of their achievable rate region in an AWGN multi-access channel with channel noise intensity equal to the lower boundary of $\mathsf{irr}(\mathcal{U})_k$, i.e. a channel where the constructing set $\mathsf{B}_k(\mathcal{U})$ achieves its maximal rate possible.

In summary, constructing the irreducible equivalent selects an AWGN multi-access channel for which the given set of user specifications corresponds to a point on the boundary of its rate achievable region, just like in the 2-user case. Again, the maximal tolerable noise intensity of the given set results directly from this construction.

We have thus completed our development on simplifying the achievability of N-user AWGN multi-access channels. Before leaving this chapter, we prove the following corollary of Theorem B.0.10 for future use.

Corollary B.0.18. *Let $\mathcal{U}$ be an overlapping set, and $\{\mathcal{U}_j, j \in [1, \dots, M]\}$ be a disjoint partition of $\mathcal{U}$. Then $\{\mathsf{T}(\mathcal{U}_j), j \in [1, \dots, M]\}$ has the overlapping property.*

Proof. Observe that combined user of any subset of the new set is the combined user of a subset in the original set, and the corollary follows Theorem B.0.10. □

Appendix C

Successive Decoding in General Discrete Memoryless Multi-Access Channels

A 3-user discrete memoryless multi-access channel illustrates the essential ideas. Let the channel be described by a channel transition probability from the joint channel inputs of the three users, $\{X_1, X_2, X_3\}$ to the channel output Y, $P(Y|X_1, X_2, X_3)$. Suppose the users transmissions are successively decodable where X_1 is decoded first, X_2 is decoded second, and X_3 is decoded last. So for decoding user 1's message, the transition probability between user 1's channel input and channel output is the marginal probability $P(Y|X_1)$. Given that user 1's message (from the first decoding stage) is decoded to be $\hat{x}_1$, the transition probability for decoding user 2's transmission is the conditional marginal probability $P(Y|X_2, X_1 = \hat{x}_1)$; and finally, given that user 2's message is decoded to be $\hat{x}_2$, the transition probability for decoding user 3's transmission is the conditional probability $P(Y|X_3, X_2 = \hat{x}_2, X_1 = \hat{x}_1)$.

Observe that only the conditional probability used to decode the last user, user 3, is taken from the given channel transition probability directly without additional computation. In general, the only known method of computing the other two transition probabilities for arbitrary discrete memoryless multi-access channels is using Bayes's rule. Specifically, let $Q(X_2)$ and $Q(X_3)$ be the probability of codewords of users 2 and 3 (assuming the messages of the users are independently chosen),

$$P(Y|X_1) = \sum_{X_2,X_3} P(Y|X_1, X_2, X_3)\, Q(X_2)Q(X_3) \tag{C.1}$$

$$P(Y|X_2, X_1 = \hat{x}_1) = \sum_{X_3} P(Y|X_1 = \hat{x}_1, X_2, X_3)\, Q(X_3) \tag{C.2}$$

where the sum is over all possible joint channel inputs (codewords combinations).

Clearly, the transition probabilities required for decoding users 1 and 2 may either be computed during the decoding process, or pre-computed and stored in additional memory. Suppose each user has a codebook of size K. First, consider computing them during the decoding process. The transition probability required for decoding the first user, as in (C.1), is the sum of K^2 terms, and that for decoding the second user, as in (C.1), is the sum of K terms. Since no more than K comparisons are necessary to determine which one of the K message was sent, we conclude that decoding the first user requires the most computations, and successive decoding in this case requires $O(K^2)$ computation.

Alternatively, we consider pre-computing these two transition probabilities. Observe that the transition probability for decoding the first user has K entries, and that for decoding the second user has K^2 entries. Hence a total of $O(K^2)$ additional memory elements are required, in addition to a total of no more than $3K$ comparisons for determining the messages from the three users.

Finally, we may combine the two approach, i.e. pre-compute and store the transition probability for decoding the first user, and compute the transition probability for decoding the second user during the decoding process. In this case, a total of K additional memory elements are required, and K terms are summed for computing the transition probability.

Generalizing to the N-user discrete memoryless multi-access channel case, we observe that N such transition probabilities are necessary for successive decoding. In particular, the transition probabilities required for decoding the first and the last users are similar to that in the 3-user case; and the transition probability required for decoding every one of the other $N-2$ users is a conditional marginal probability similar to that for decoding the middle user in the 3-user case. Again, these transition probabilities may be either computed during the decoding process, or pre-computed and stored in memory. Suppose each user has a codebook of size K. We conclude that computing these transition probabilities during decoding requires a total of $O(K^{N-1})$ computation; pre-computing and storing them requires a total of $O(K^{N-1})$ additional storage elements; and combining the two approach requires a total of $O(K^{\lfloor N/2 \rfloor})$ computation and $O(K^{\lfloor N/2 \rfloor})$ additional storage elements.

The brute-force joint decoding regards the multi-access channel as one single-user channel, where the messages from all the users are combined into one. Thus, the decoding chooses one of the K^N possible joint codewords. It therefore has an $O(K^N)$ computation burden.

In comparison, computing all the transition probabilities during successive decoding does not result in substantial saving. As for pre-computing and storing these transition probabilities, it is difficult to say how to compare the additional memory requirement to computation burdens, particularly since the previously decoded user messages need be decoded reliably in order for the next user's message to be decoded reliably using the successive decoding, and that reliable communication through a noisy channel may only be achieved as the number of channel symbols transmitted and the size of codebook K grow to infinity.

Appendix D

The Non-Optimality of the Gaussian Input Assumption for Successive Decoding

Theorem D.0.19. *For successive decoding, the theoretical Gaussian transmission codebooks are not optimal. Specifically, for given user instantaneous transmission power constraints, there are rate tuples outside the direct successive decodable region with the constraint that all channel inputs be independent Gaussian that are directly successive decodable with non-Gaussian channel inputs.*

Proof. We consider a two-user multi-access channel.

Let Z be the additive white Gaussian noise with intensity n_o. Let X_1 and X_2 be the channel input of users 1 and 2 respectively. Moreover, the two users observe instantaneous transmission power constraints p_1 and p_2 respectively.

At the other end of the channel, the transmission of user 2 is to be decoded first then subtracted. Then the transmission of user 1 is to be decoded without any interference from the transmission of user 2.

We are interested in maximizing the combined transmission rate of the two users under these transmission and decoding order constraints.

From the previous discussions, since user 2 is to be decoded first regarding the transmission of user 1 as noise, user 2 should certainly transmit at maximal power; whereas it may be desirable to have user 1 transmit with power lower than its maximum.

To simplify our notation, let

$$Y_1 = Z + X_1 \tag{D.1}$$

$$Y_2 = Y_1 + X_2 = Z + X_1 + X_2 \tag{D.2}$$

Let σ_1 and σ_2 denote the instantaneous transmission power constraint of users 1 and 2 respectively. There is

$$\sigma_1 \leq p_1 \tag{D.3}$$

$$\sigma_2 = p_2 \tag{D.4}$$

For given distribution of the two users' channel inputs, let R_1 and R_2 be the best transmission rate of users 1 and 2 respectively. By definition of mutual information, there are:

$$R_1 = \mathsf{I}(Y_1 : X_1) = \mathsf{H}(Z + X_1) - \mathsf{H}(Z) \tag{D.5}$$

$$R_2 = \mathsf{I}(Y_2 : X_2) = \mathsf{H}(Z + X_1 + X_2) - \mathsf{H}(Z + X_1) \tag{D.6}$$

One way to view the optimizing problem is as maximizing R_2 for fixed $R_1 < \frac{1}{2}\ln\left(\frac{n_o+p_1}{n_o}\right)$ over all possible distribution of X_1 and X_2 that satisfy their power constraint.

Actually, all that needs to be done to prove this theorem is to show that for fixed R_1, Gaussian X_1 and X_2 do not achieve the optimal R_2. We accomplish this by contradiction.

Suppose the contrary, i.e. for fixed R_1 in the specified range, Gaussian X_1 and X_2 do achieve the optimal R_2.

Let X_1^* and X_2^* denote the Gaussian channel input of the two users that achieves the optimal R_2 for the fixed R_1. Let σ_1^* and σ_2^* denote the variance of X_1^* and X_2^* respectively. There are

$$\sigma_1^* < p_1 \tag{D.7}$$

$$\sigma_2^* = p_2 \tag{D.8}$$

And the maximal R_2 for this R_1, call it R_2^*, is given by

$$R_2^* = \mathsf{H}(Z + X_1 + X_2) - \mathsf{H}(Z + X_1) = \mathsf{H}(Z + X_1^* + X_2^*) - R_1 + \mathsf{H}(Z) \tag{D.9}$$

Now, since $\sigma_1^* < p_1$, there exists a distribution of X_1, with variance greater than σ_1^* but less or equal to p_1, such that

$$\mathsf{H}(Z + X_1^*) = \mathsf{H}(Z + X_1) \tag{D.10}$$

By the entropy power inequality,

$$\begin{aligned} e^{2\mathsf{H}(Z+X_1+X_2^*)} &> e^{2\mathsf{H}(Z+X_1)} + e^{2\mathsf{H}(X_2^*)} \\ &= e^{2\mathsf{H}(Z+X_1^*)} + e^{2\mathsf{H}(X_2^*)} \\ &= e^{2\mathsf{H}(Z+X_1^*+X_2^*)} \end{aligned} \tag{D.11}$$

By the monotonicity of the exponential function, there is

$$\mathsf{H}(Z + X_1 + X_2^*) \geq \mathsf{H}(Z + X_1^* + X_2^*) \tag{D.12}$$

Hence, the achieved R_2 in this case is given by

$$\begin{aligned} R_2 &= \mathsf{H}(Z + X_1 + X_2^*) - \mathsf{H}(Z + X_1) \\ &= \mathsf{H}(Z + X_1 + X_2^*) - \mathsf{H}(Z + X_1^*) \\ &\geq \mathsf{H}(Z + X_1^* + X_2^*) - R_1 + \mathsf{H}(Z) = R_2^* \end{aligned} \tag{D.13}$$

This contradicts our assumption and completes the proof. □

Since real communication systems do not use a Gaussian codebook, we may be able to take advantage of this situation.

Appendix E

Simplifying the Achievability in the Time-Shared AWGN Multi-Access Channel

In this appendix, we present a simplification of achievability in an N-user time-shared AWGN multi-access channel to $O(N \ln N)$.

Definition E.0.9. A *time-shared AWGN multi-access channel* is a composite channel in which a set of parallel and independent AWGN multi-access channels are time-shared according to fixed time portions. Members of the parallel and independent channels are called *sub-channels*.

Here, we use the term *time-sharing* exactly as discussed in Chapter 5. Specifically,

1. the receiver and all transmitters in the system have the knowledge of which sub-channel is used for each transmission, and
2. each transmitter is required to observe its power constraint regardless of which sub-channel is being used

In the following development, we assume the number of sub-channels in the system is finite[1]. Figure E-1 illustrates this channel model.

For $j \in [1, \dots, J]$, let n_j denote the noise variance in sub-channel j, and λ_j denote the portion of time that channel j is used. Following the description of time-sharing, one may model the effect of this (discrete time) time-shared AWGN multi-access channel as follows:

1. Section all channel uses into blocks each consists of K channel uses

[1]The generalizations of our results to the case of arbitrary number of sub-channels are straight-forward.

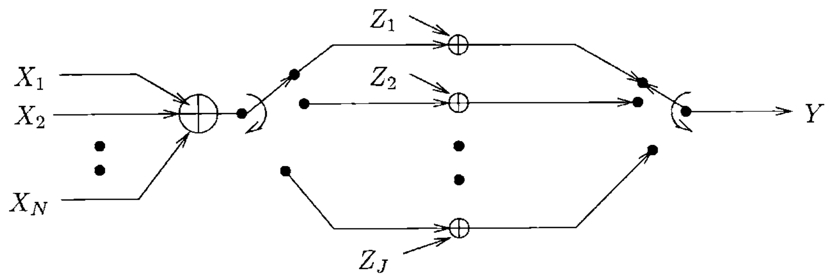


Figure E-1: The time-shared AWGN multi-access channel channel model

2. Within each K-block, channel j is used between channel uses l_{i-1} and $l_i - 1$, where

$$l_0 = 0 \tag{E.1}$$

$$l_j = \left\lfloor K \times \sum_{j'=1}^{j} \lambda_j \right\rfloor, j \in [1, \dots, J] \tag{E.2}$$

In other words, the noise variance for channel uses l_{i-1} through $l_i - 1$ is n_j.

The second constraint, which states that each transmitter must devote identical amount of power to the transmissions in each sub-channel, affects the capacity of this channel. We consider the capacity of the time-shared AWGN multi-access channel next.

E.1 The Achievability of the Time-Shared AWGN multi-access Channel

We start with the case of two sub-channels and a single user (transmitter) because the discussions on the achievable rate region for this case embodies the essential elements for the general case.

Let p and r denote the power constraint and the average transmission rate (per channel use or per transmission) of the user in this channel. We now show that r is achievable in this channel if and only if r is no greater than

$$\frac{\lambda_1}{2} \ln\left(\frac{n_1 + p}{n_1}\right) + \frac{\lambda_2}{2} \ln\left(\frac{n_2 + p}{n_2}\right) = \lambda_1 \mathsf{C}(p, n_1) + \lambda_2 \mathsf{C}(p, n_2) \tag{E.3}$$

To see this, recall that the channel capacity of the j^{th} sub-channel is $\mathsf{C}(p, n_j)$ per channel use, where $j \in [1, 2]$. By definition of time portion, this sub-channel is used, on average, λ_j times per transmission in the time-shared AWGN multi-access channel. Therefore an average rate of no greater than $\lambda_j \mathsf{C}(p, n_j)$ per channel use may be reliably communicated through this sub-channel. Hence the average mutual information of this time-shared AWGN multi-access channel may not exceed the sum of maximal average rates in both sub-channels, which is expressed in (E.3), in units of per transmission. This shows that no rate greater than (E.3) per transmission may be reliably transmitted through this time-shared AWGN multi-access channel.

Note that having $\lambda_j = 0$ for some j implies channel j is not used[2]. For this reason, we may assume all λ_j concerned in the computation of achievability in the rest of this section to be non-zero.

Consider this user's transmissions through sub-channel j only. Let

$$\gamma_j < \mathsf{C}(p, n_j) = \frac{1}{2} \ln \left(\frac{n_j + p}{n_j} \right) \tag{E.4}$$

By definition of channel capacity, a information rate of γ_j (in units of per channel use) may be reliably communicated through sub-channel j with a transmission codebook $\mathcal{C}_j$ and a positive error exponent $\mathcal{E}_j$ (in units of per channel use).

The utility portion of this sub-channel in a time-shared AWGN multi-acccss channel may be modeled as using this sub-channel, on average, once every $\frac{1}{\lambda_j}$ channel uses in the time-shared AWGN multi-access channel. Hence, the codebook $\mathcal{C}_j$ may be used during sub-channel j to reliably transmit at the rate $\lambda_j \gamma_j$ with an (average) error exponent $\lambda_j \mathcal{E}_j$, both in units of per channel use (in the time-shared AWGN multi-access channel). Observe that since $\mathcal{E}_j > 0$, so must $\lambda_j \mathcal{E}_j$.

By decoding the transmissions in the both sub-channels independently and then combining the decoding results, a information rate $\lambda_1 \gamma_1 + \lambda_2 \gamma_2$ may be achieved in the time-shared AWGN multi-access channel using $\mathcal{C}_1$ and $\mathcal{C}_2$ with an error exponent lower bounded by $\min_{j \in [1,2]} \lambda_j \mathcal{E}_j$ per channel use (in the time-shared AWGN multi-access channel). Since both operands in the minimization are positive, so must their minimum. In other words, the information rate $\lambda_1 \gamma_1 + \lambda_2 \gamma_2$ can be reliably transmitted through this time-shared AWGN multi-access channel. Since the two γ_j's are arbitrarily chosen to satisfy (E.4), we conclude that every rate upto the expression in (E.3) may be reliably communicated through this time-shared AWGN multi-access channel. This establishes (E.4) to be the capacity of this channel.

Note that we showed that every rate upto the capacity may be achieved by independent coding and decoding transmissions in each of the sub-channels. We note that even though coding the transmissions through the two channels together does not enlarge the capacity of this channel, it can be used to improve the overall error exponent.

[2]To be precise, $\lambda_j = 0$ implies channel j is not used sufficiently frequent to contribute to the transmission rate of any user in the system.

We now generalize the arguments above to a time-shared AWGN multi-access channel consisting of N users (transmitters).

For each $i \in [1, \dots, N]$, let p_i and r_i respectively denote the power constraint and transmission rate specifications of user i.

Fixing $j \in \{1, 2\}$, consider transmissions through only sub-channel j. Let

$$\mathsf{R}_j = (\gamma_{1,j}, \gamma_{2,j}, \cdots, \gamma_{N,j})$$

denote a rate tuple at which the N users may reliably communicate with the receiver in the j^{th} sub-channel. Since, by definition of time-shared AWGN multi-access channel, every user must transmit with identical power constraints in every sub-channel, we have

$$\sum_{i \in \mathcal{S}} \gamma_{i,j} < \mathsf{C}\left(\sum_{i \in \mathcal{S}} p_i, n_j\right) = \frac{1}{2} \ln\left(1 + \frac{\sum_{i \in \mathcal{S}} p_i}{n_j}\right), \forall \mathcal{S} \subseteq [1, \dots, N] \tag{E.5}$$

By definition of reliable communication, there exists a set of transmission codebooks, one for each user, such that the probability of any decoding error when transmitting at R_j decreases exponentially with the length of transmission codebook in this sub-channel. For each $i \in [1, \dots, N]$, we use $\mathcal{C}_{i,j}$ to denote user i's transmission codebook (in this sub-channel). We use $\mathcal{E}_j$ to denote this error exponent.

Note that the same dividing and re-combining technique used in the single-user case above can be repeated for each users' messages to achieve the rate tuple $\lambda_1 \mathsf{R}_1 + \lambda_2 \mathsf{R}_2$ in the time-shared AWGN multi-access channel. Specifically, for each $i \in [1, \dots, N]$, divide user i's message into two pieces, one with average rate $\lambda_1 \gamma_{i,1}$, the other with average rate $\lambda_2 \gamma_i 2$. Then instruct user i to transmit the first piece only in the first sub-channel (the one with noise variance n_1) using codebook $\mathcal{C}_{i,1}$, and the second piece only in the second sub-channel using codebook $\mathcal{C}_{i,2}$. At the receiver, the transmissions received in each sub-channel is first decoded, and the decoded messages are then re-combined to recover the original messages of each user.

Observe that the error exponent (for any decoding error) in the resulting combined transmission is lower bounded by $\min_{j \in [1,2]} \lambda_j \mathcal{E}_j$ per channel use in the time-shared AWGN multi-access channel. Since both terms are positive, so must their minimum. Hence the N users are able to reliably communicate with the receiver at the rate tuple $\lambda_1 \mathsf{R}_1 + \lambda_2 \mathsf{R}_2$ in this time-shared AWGN multi-access channel. Since R_1 and R_2 are arbitrarily chosen to satisfy (E.5) for their respective j. We conclude a rate tuple $\mathsf{R} = (r_1, r_2, \dots, r_N)$ is achievable in this channel if

$$\sum_{i \in \mathcal{S}} r_i \le \lambda_1 \mathsf{C}\left(\sum_{i \in \mathcal{S}} p_i, n_1\right) + \lambda_2 \mathsf{C}\left(\sum_{i \in \mathcal{S}} p_i, n_2\right), \ \forall \mathcal{S} \subseteq [1, \dots, N] \tag{E.6}$$

The converse, which states that no other rate tuple may be reliably communicated through this time-shared AWGN multi-access channel, can be argued use the interpretation on the AWGN multi-access achievable rate region offered by Gallager in [8]. The interpretation is as follows: suppose a super-user is able to coordinate the transmission of all users in a subset $\mathcal{S}$, and tell the remaining users to remain silent, then the transmission rate achieved by the super-user may not be exceeded by the combined individual efforts of all users in $\mathcal{S}$. This interpretation can be used, in conjunction of the capacity of time-shared AWGN multi-access channel with single-user above, to establish each conditions in (E.6) in turn. We have thus shown that a rate tuple R is achievable in this time-shared AWGN multi-access channel if and only if it satisfies all the conditions in (E.6).

Observe that increasing the number of sub-channels in the system only increases the number of piece in the dividing and re-combining technique used in the argument. The generalization to the J arbitrary sub-channels is therefore trivial. We therefore have the following.

Theorem E.1.1. *For each $j \in [1, \dots, J]$, let n_j and λ_j respectively denote the channel noise variance and the utility portion of the j^{th} sub-channel. For each $i \in [1, \dots, N]$, let p_i and r_i respectively denote the power constraint and the (average) transmission rate of user i. We conclude that a rate tuple R is achievable in this time-shared AWGN multi-access channel if and only if*

$$\sum_{i \in \mathcal{S}} r_i \leq \sum_{j=1}^{J} \lambda_j \mathsf{C}\left(\sum_{i \in \mathcal{S}} p_i, n_j\right), \ \forall \mathcal{S} \subseteq [1, \dots, N] \tag{E.7}$$

For simplicity, we will refer to (E.7) as the set of *time-shared AWGN multi-access achievability conditions.*

As noted in the single user and two sub-channels case above, we have argued that coding and decoding a user's transmissions in each sub-channel independently does not affect the capacity region. However, coding a user's transmissions through all channels together may improve the resulting decoding error exponent.

Observe that (E.7) consists of $O(2^N)$ conditions, each corresponding to a unique subset $\mathcal{S}$ of $[1, \dots, N]$. In particular, the time-shared AWGN multi-access achievability have identical structure as the AWGN multi-access achievability. In the next section, we will extend the power diagram framework, and develop an algorithm that verifies whether a given set of user specifications is achievable in a time-shared AWGN multi-access channel requiring $O(N \ln N)$ computations.

E.2 Simplifying the Time-Shared AWGN Multi-Access Achievability with the Generalized Power Diagram

In this section, we simplify the N-user time-shared AWGN multi-access achievability conditions established in the previous section to an algorithm that requires $O(N \ln N)$ operations. Both the approach and the constructions used in this simplification mimic those used in simplifying the AWGN multi-access achievability in Chapter 2.

Recall that our simplification of the AWGN multi-access achievability is facilitated by the new perspective resulting from the new associations of the user and channel attributes; and the actual simplification is accomplished using the power diagram, which provides a convenient graphical operational framework that embodies the necessary details of the new perspective.

The simplification of the time-shared AWGN multi-access achievability also takes place on these two levels. In particular, the attributes of a user in a time-shared AWGN multi-access channel are still the power constraint and desired transmission rate. The channel attribute is generalized to include the noise variances and time portions of all sub-channels in the system. Hence, in a time-shared AWGN multi-access channel consisting of J sub-channels, the channel specification consists J pairs of noise variances and the time portions.

For each $j \in [1, \dots, J]$, let n_j and λ_j respectively denote the channel noise variance and the utility portion of the j^{th} sub-channel for each $j \in [1, \dots, J]$. For simplicity of notation, we use the pair (n_j, λ_j) to denote the specification of the j^{th} sub-channel, and abbreviate the specifications of this time-shared AWGN multi-access channel as $\mathsf{n} = \{(n_j, \lambda_j), j \in [1, \dots, J]\}$.

In the following, we first generalize the power interval representation of user specifications to time-shared AWGN multi-access channel. With the appropriately generalized power interval representations, the desired simplification exactly parallels that of AWGN multi-access channels.

Definition E.2.1. Let (p, r) denote the specifications of a user in this time-shared AWGN multi-access channel. Then this user's *generalized power interval* (in this channel) is the left-open-right-closed interval $(\eta, \eta + p]$ on the real axis, where η satisfy

$$r = \sum_{j=1}^{J} \lambda_j \mathsf{C}\left(p, n_j + \eta\right) \tag{E.8}$$

The real axis on which generalized power intervals in the same channel are placed is called the *generalized power axis* (for this channel).

In our discussion below, when it is necessary to distinguish generalized power intervals in different time-shared AWGN multi-access channels, we will employ the notation $(\eta, \eta + p]_{\mathsf{n}}$ to denote the generalized power interval in a time-shared AWGN multi-access channel with

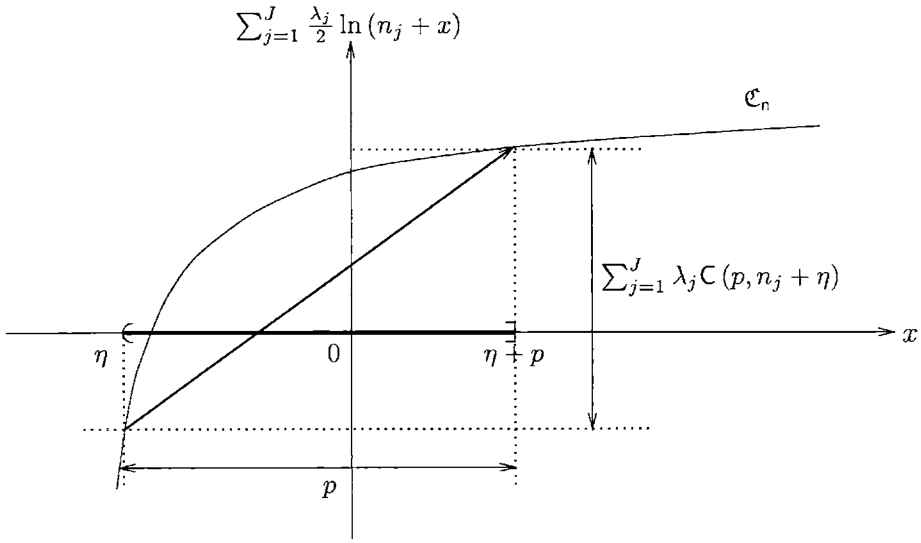


Figure E-2: Interpreting the generalized power interval representation

specifications n. However, when the context of discussion makes it clear which is the concerned time-shared AWGN multi-access channel, we may omit the subscript for notational simplicity.

Observe that the generalized power interval of a user is a left-open-right-closed interval on the real axis, just like a user's power interval (see Definition 2.1.1). Additionally, the length of a user's generalized power interval is the power constraint of the user, just like that of a power interval. In fact, the only difference between the two is in the equation that their lower boundaries, η, must satisfy.

The physical significance of this generalized power interval may be seen from a graphical interpretation just like the one used for power intervals. To understand this, notice that from (E.8), we have

$$r = \sum_{j=1}^{J} \lambda_j \mathsf{C}(p, n_j + \eta) \tag{E.9}$$

$$= \sum_{j=1}^{J} \lambda_j \mathsf{r}((n_j + \eta, n_j + \eta + p]) \tag{E.10}$$

$$= \sum_{j=1}^{J} \frac{\lambda_j}{2} \ln(n_j + \eta + p) - \sum_{j=1}^{J} \frac{\lambda_j}{2} \ln(n_j + \eta) \tag{E.11}$$

In other words, the rate specification of this user (and hence this user's generalized power interval) may be obtained from the height difference which the function $\sum_{j=1}^{J} \frac{\lambda_j}{2} \ln(n_j + x)$ achieves over the horizontal interval $(\eta, \eta + p]$. This is illustrated in Figure E-2.

From this figure, observe that the generalized power interval of this user may be obtained by first constructing a vector with length and height that are respectively equal to the power and rate specifications of the user, then placing both ends of this vector on the $\sum_{j=1}^{J} \frac{\lambda_j}{2} \ln(n_j + x)$ curve. The projection of the placement of this vector onto the horizontal axis is the generalized power interval of the user. To facilitate comparison, recall that the same location and projection operations can be performed on $\mathfrak{C}$ to obtain a user's power interval.

For simplicity, we will use $\mathfrak{C}_n$ to denote the curve for $\sum_{j=1}^{J} \frac{\lambda_j}{2} \ln(n_j + x)$. Observe that since $\frac{1}{2} \ln(n_j + x)$ is continuous, monotonic and concave down (cap) for $x > -n_j$ for each j, the same must be true for $\mathfrak{C}_n$ for $x > -\min_{j \in [1,\ldots,N]} n_j$. Note that these are the same properties that $\mathfrak{C}$ has, with the only difference being that $\mathfrak{C}$ is defined for $x > 0$. In addition, since all results concerning properties of sets of power intervals developed in the previous chapters are based only on the continuity, monotonicity, and the concavity of $\mathfrak{C}$, we would expect that these results holds true with sets of generalized power intervals (when the constructions involved are appropriately extended). This is our approach to simplifying the time-shared AWGN multi-access achievability, which we will present below.

For the moment, we develop a few more insights into the generalized power intervals. Without the loss of generality, let

$$n_1 \leq n_2 \leq \ldots \leq n_J \tag{E.12}$$

Note that

$$\lim_{x \to -n_1} \frac{\lambda_1}{2} \ln(n_1 + x) = -\infty \tag{E.13}$$

Therefore,

$$\lim_{x \to -n_1} \sum_{j=1}^{J} \frac{\lambda_j}{2} \ln(n_j + x) = -\infty \tag{E.14}$$

Combined with the monotonicity and concavity of $\mathfrak{C}_n$, we conclude that the placement of a vector with arbitrary (positive) length and height on this curve must be unique. In other words, the generalized power interval representation of a user's specifications is unique. Note also that a user's generalized power interval may contain elements from the negative part of the real axis.

Now, recall that the lower boundary of a power interval is the maximal noise threshold in an AWGN multi-access channel for the user's specifications to be achievable. The lower boundary of a generalized power interval has a similar interpretation. To see this, first recall, from the previous discussion, that a user with specification (p, r) is a achievable in this time-shared

AWGN multi-access channel if and only if

$$r \leq \sum_{j=1}^{J} \lambda_j \mathsf{C}(p, n_j) \tag{E.15}$$

Let $(\eta, \eta + p]$ denote the generalized power interval of the user in this time-shared AWGN multi-access channel. By definition, we have

$$\sum_{j=1}^{J} \lambda_j \mathsf{C}(p, n_j + \eta) \leq \sum_{j=1}^{J} \lambda_j \mathsf{C}(p, n_j) \tag{E.16}$$

By the monotonicity and concavity of $\mathfrak{C}_{\mathsf{n}}$, this condition is equivalent to

$$0 \leq \eta \tag{E.17}$$

In other words, we conclude that the user's specifications is achievable in this time-shared AWGN multi-access channel if and only if the lower boundary of the user's generalized power interval is non-negative.

To better understand the significance of the lower boundary of a generalized power interval, consider interpreting generalized power intervals and channel specifications of time-shared AWGN multi-access channel in the time-power diagram. To start, note that we have modeled time-sharing of channels in a time-shared AWGN multi-access channel exactly following time-sharing among sets of user specifications. Hence, we may construct a power diagram to include both the noise intensity and the user's power and rate specifications for each channel use, which results in a time-power diagram following the same construction in Section 5.1. Let $a_0 = 0$ and $a_j = \sum_{j'=1}^{j} \lambda_j$ for each $j \in [1, \dots, J]$. We may therefore represent the channel specifications n in the time-power diagram as the set of line segments

$$\{n_j \times (a_{j-1}, a_j], j \in [1, \dots, J]\} \tag{E.18}$$

Note that each sub-channels in the system may be represented by a coherent segment.

The user's power interval in each coherent segment may be obtainted from the second equivalence in (E.11). This equivalence states that the rate specification of the generalized power interval $(\eta, \eta + p]_{\mathsf{n}}$ is equivalent to the sum of rates of power intervals $(n_j + \eta, n_j + \eta + p]$ weighed by λ_j over all $j \in [1, \dots, J]$. Since the user is constrained to transmit with identical amount of power at all times, we may regard $(n_j + \eta, n_j + \eta + p]$ to be user's power interval in sub-channel j. This user can thus be represented by the simple set of time-power bi-intervals

$$\{(n_j + \eta, n_j + \eta + p] \times (a_{j-1}, a_j], j \in [1, \dots, J]\} \tag{E.19}$$

Note that this set of time-power bi-intervals is an achieving set of a user.

We may therefore regard the set of time-power bi-intervals in (E.19) as the representation of the generalized power interval $(\eta, \eta + p]_{\mathsf{n}}$ in the time-power diagram, and regard $(\eta, \eta + p]_{\mathsf{n}}$ as the generalized power interval representation of the set of time-power bi-intervals in (E.19).

Comparing the representations of the channel and user in the time-power diagram, i.e. (E.18) and (E.19), note that η specifies the common distance between the noise variance and the lower boundary of the user's power interval in each of the coherent segments. Note that this representation of the user's specification as a set of time-power intervals with uniform distance from the noise variances in their respective coherent segment (sub-channels) reflects on the fact that a user in the time-shared AWGN multi-access channel is constrained to transmit with identical amount of power at all times (rather, to leave identical amount of power for the user to be decoded next). From this interpretation, the understanding of achievability condition in (E.17) follows immediately.

Consider a second time-shared AWGN multi-access channel with channel specification $\mathsf{n}' = \{(n_j - c, \lambda_j), j \in [1, \dots, J]\}$ for some constant c. The representation of this channel specification in the time-power diagram is

$$\{(n_j - c) \times (a_{j-1}, a_j], j \in [1, \dots, J]\} \tag{E.20}$$

Compared to the representation of n in the time-power diagram, note that the line segments of the two are exactly c units apart in every coherent segment. Let $(\eta', \eta' + p]_{\mathsf{n}'}$ denote the generalized power interval of the above user in the new channel, and let $\widehat{\mathcal{V}}$ denote its representation in the time-power diagram. Recall that each time-power interval in $\widehat{\mathcal{V}}$ must be placed at a uniform distance from the channel noise variances in their respective coherent segment. Hence they must be placed at a uniform distance from the channel noise variances of n in their respective coherent segment. Finally, since $(\eta', \eta' + p]_{\mathsf{n}'}$ has identical power and rate specifications as $(\eta, \eta + p]_{\mathsf{n}'}$, we conclude that their representation in the time-power diagram must be identical, i.e.

$$\eta' = \eta + c \tag{E.21}$$

In other words, generalized power interval is translational invariant with regard to uniform variation of noise variances of all sub-channels in the system. Note that these conclusions may also be reached using the vector interpretation on the $\mathfrak{C}_n$ curve by noting that altering the noise variances of all sub-channels uniformly corresponds to linearly shifting the generalized power axis.

Since the generalized power interval of every user in a given time-shared AWGN multi-access channel (or a class of time-shared AWGN multi-access channel as discussed above) are derived using a single $\mathfrak{C}_n$ curve, we therefore may consider placing multiple generalized power intervals on the same generalized power axis. In the following, we refer to such a construct as the *generalized power diagram.*

Recall that the new perspective in Chapter 2 made it possible to consider properties of sets of user specifications independent of the channel attributes (noise variance). This is also possible in a time-shared AWGN multi-access channel using the associations above. To illustrate, we consider the property of a set of user specifications in a time-shared AWGN multi-access channel that is analogous to the maximal noise threshold in AWGN multi-access channels. From the discussions above, recall that the lower boundary of a generalized power interval indicates the (common) maximal amount that the noise variances of all the sub-channels may be raised without violating the achievability of the user's specifications in the resulting channel. Extending this understanding, we consider the question

> what is the (common) maximal amount that the noise variances of all the sub-channels may be raised without violating the achievability of the set of user specifications in the resulting channel?

As in the case of AWGN multi-access channels, the solution to this question may be obtained from reformulating the time-shared AWGN multi-access achievability conditions using the new associations of attributes and the generalized power intervals.

For convenience, we respectively extend the definition of $\eta(\cdot)$ and $\mathsf{p}(\cdot)$ to generalized power interval, and generalize the definition of $\mathsf{r}(\cdot)$ in the following.

Definition E.2.2. Given a user's generalized power interval $T = (a, b]$ in a time-shared AWGN multi-access channel with channel specifications $\mathsf{n} = \{(n_j, \lambda_j), j \in [1, \dots, J]\}$,

- $\eta(T) \equiv a$ is the lower boundary of the generalized power interval, which, by definition, is also the maximal amount that the noise variances of all the sub-channels may be raised without violating the achievability of this user in the resulting time-shared AWGN multi-access channel.

- $\mathsf{p}(T) \equiv \mathsf{p}\left((a, b]\right) = b - a$ is the length of the interval, which is the user's power constraint.

- $\mathsf{r_n}(T) \equiv \sum_{j=1}^{J} \lambda_j \mathsf{r}((n_j + a, n_j + b]) = \sum_{j=1}^{J} \lambda_j \mathsf{C}\left(\sum_{i \in \mathcal{S}} p_i, n_j + \eta_{\mathcal{S}}\right)$ is the rate specification of the generalized power interval in this channel.

Recall from previous discussions, that a rate tuple R is achievable in this time-shared AWGN multi-access channel if and only if

$$\sum_{i \in \mathcal{S}} r_i \leq \sum_{j=1}^{J} \lambda_j \mathsf{C}\left(\sum_{i \in \mathcal{S}} p_i, n_j\right), \ \forall \mathcal{S} \subseteq [1, \dots, N] \tag{E.22}$$

As in reformulating the AWGN multi-access achievability using the new associations of attributes in Chapter 2, we consider constructing, for each $\mathcal{S} \in [1, \dots, N]$, the combined user of the set of users with indices in $\mathcal{S}$. By definition, this combined user is specified by

$$\left(\sum_{i \in \mathcal{S}} p_i, \sum_{i \in \mathcal{S}} r_i\right) \tag{E.23}$$

Let $\left(\eta_{\mathcal{S}}, \eta_{\mathcal{S}} + \sum_{i \in \mathcal{S}} p_i\right]$ denote its generalized power interval in this time-shared AWGN multi-access channel. By definition of generalized power intervals,

$$\begin{aligned}\sum_{i \in \mathcal{S}} p_i &= \mathsf{r_n}\left(\left(\eta_{\mathcal{S}}, \eta_{\mathcal{S}} + \sum_{i \in \mathcal{S}} p_i\right]\right) \\ &= \sum_{j=1}^{J} \lambda_j \mathsf{C}\left(\sum_{i \in \mathcal{S}} p_i, n_j + \eta_{\mathcal{S}}\right)\end{aligned} \tag{E.24}$$

From discussions on the generalized power intervals, e.g. (E.17), we know that this combined user is achievable in this time-shared AWGN multi-access channel if and only if

$$0 \leq \eta_{\mathcal{S}} \tag{E.25}$$

Therefore, the time-shared AWGN multi-access achievability is equivalent to

$$0 \leq \min_{\mathcal{S} \subseteq [1, \dots, N]} \eta_{\mathcal{S}} \tag{E.26}$$

We conclude that the right-hand-side of (E.26) gives the desired maximal amount quantity. As note above, the channel attribute is abstracted to be 0.

This formulation states that the simplification the time-shared AWGN multi-access achievability may be accomplished by simplifying the minimization of the lower boundaries of the set of corresponding (combined) generalized power intervals. For comparison, recall that we observed in the introduction of Chapter 2 that the simplification of the AWGN multi-access achievability may be accomplished by simplifying the minimization of the lower boundaries of the set of corresponding (combined) power intervals.

As noted earlier, $\mathfrak{C}_{\mathsf{n}}$ is continuous, monotonic, and concave just like $\mathfrak{C}$. Since all properties of power intervals and power interval constructs developed in previous chapters require only these properties, we can extend these properties and constructs for the generalized power intervals.

Recall that in Chapter 2, we simplified the AWGN multi-access achievability using the irreducible equivalent of the set of user power intervals. In the above, we have just shown that the time-shared AWGN multi-access achievability has the same formulation under the new association of attributes as the one for the AWGN multi-access achievability we started with in Chapter 2, we would expect that suitably generalizing the irreducible equivalent construction for generalized power intervals will lead to similar simplifications of the time-shared AWGN multi-access achievability.

In the following, we present this generalization paralleling the simplification of the AWGN multi-access achievability. As will become clear shortly, many results may be proven following identical logic as those used in proving as their counterparts in Chapter 2. For simplicity, we will omit the details of some of these proofs below.

First, we define the following notation.

Definition E.2.3. Given a set of user specifications $\mathcal{U}$. We denote the generalized power interval of their combined user in a time-shared AWGN multi-access channel with specifications n as $\mathsf{T}_{\mathsf{n}}(\mathcal{U})$.

Consider the following generalization of Lemma 2.2.2.

Lemma E.2.1. *Let n denote the specification of a time-shared AWGN multi-access channel, and T_1 and T_2 be two generalized power intervals in this channel. Suppose T_1 and T_2 are adjacent, then the generalized power interval of their combined user in this channel is the union of the two, i.e.* $\mathsf{T}_{\mathsf{n}}(\{T_1, T_2\}) = T_1 \cup T_2$.

Remark. Figure E-3 illustrates the result of this lemma using the vector representation of generalized power intervals.

Proof. Let $T_1 = (a_1, b_1]$ and $T_2 = (a_2, b_2]$. By assumption, T_1 and T_2 are adjacent. Without the loss of generality, let

$$a_2 = b_1 \tag{E.27}$$

Let $T = (a, b] = \mathsf{T}_{\mathsf{n}}(\{T_1, T_2\})$.

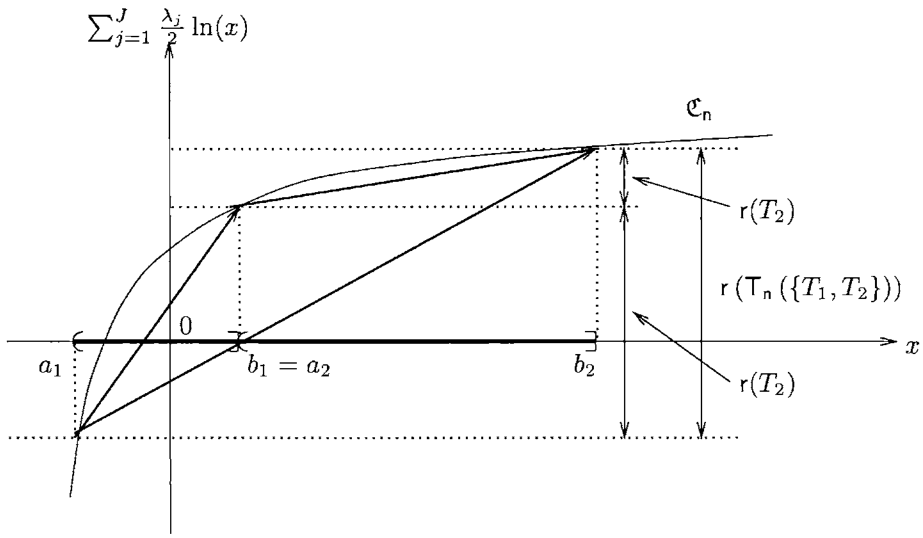


Figure E-3: The addition of two generalized power interval and their vector representations

By definition of the combined user,

$$\mathsf{p}(T) = b - a = \mathsf{p}(T_1) + \mathsf{p}(T_2) = (b_1 - a_1) + (b_2 - a_2) \tag{E.28}$$

$$\mathsf{r_n}(T) = \sum_{j=1}^{J} \lambda_j \mathsf{r}\left((n_j + a, n_j + b]\right) = \mathsf{r_n}(T_1) + \mathsf{r_n}(T_2) \tag{E.29}$$

By definition,

$$\mathsf{r_n}(T_1) = \sum_{j=1}^{J} \lambda_j \mathsf{r}\left((n_j + a_1, n_j + b_1]\right) \tag{E.30}$$

$$\mathsf{r_n}(T_2) = \sum_{j=1}^{J} \lambda_j \mathsf{r}\left((n_j + a_2, n_j + b_2]\right) \tag{E.31}$$

Summing these two equations, we have the following on the rate specification of T:

$$\begin{aligned}
\mathsf{r_n}(T_1) + \mathsf{r_n}(T_2) &= \sum_{j=1}^{J} \lambda_j \mathsf{r}\left((n_j + a_1, n_j + b_1]\right) + \sum_{j=1}^{J} \lambda_j \mathsf{r}\left((n_j + a_2, n_j + b_2]\right) \\
&= \sum_{j=1}^{J} \lambda_j \left(\mathsf{r}\left(((n_j + a_1, n_j + b_1]\right) + \mathsf{r}\left((n_j + a_2, n_j + b_2]\right)\right)\right) \\
&= \sum_{j=1}^{J} \lambda_j \left(\mathsf{r}\left(((n_j + a_1, n_j + b_1]\right) + \mathsf{r}\left((n_j + b_1, n_j + b_2]\right)\right)\right) \\
&= \sum_{j=1}^{J} \lambda_j \left(\mathsf{r}\left((n_j + a_1, n_j + b_2]\right)\right)
\end{aligned} \tag{E.32}$$

The last two equalities are by (E.27) and Lemma 2.2.2 respectively.

Combining with (E.29), we have

$$\mathsf{r}_{\mathsf{n}}(T) = \sum_{j=1}^{J} \lambda_j \left(\mathsf{r}\left((n_j + a_1, n_j + b_2]\right)\right) \tag{E.33}$$

The conclusion of this lemma follows (E.28) and the uniqueness of generalized power intervals discussed above. □

We now generalize the results in Theorem 2.2.1.

Theorem E.2.2. *Let* n *denote the specification of a time-shared AWGN multi-access channel,* T_1 *and* T_2 *be two generalized power intervals in this channel, and* T *be the generalized power interval of the two's combined user in this channel, i.e.* $T = \mathsf{T}_{\mathsf{n}}(\{T_1, T_2\})$. *Then*

- *if* $T_1 \cap T_2 \neq \emptyset$, *the closure of* $\mathsf{ext}(\{T_1, T_2\})$ *is contained in the interior of* T, *i.e.*

$$\overline{\mathsf{ext}(\{T_1, T_2\})} \subset T^o$$

- *if* T_1 *and* T_2 *are adjacent, their combined user is equal to both their extent and their union, i.e.* $T = \mathsf{ext}(\{T_1, T_2\}) = T_1 \cup T_2$.
- *otherwise,* T_1 *and* T_2 *are separated, and* $\overline{T} \subset \mathsf{ext}\left(\{T_1, T_2\}\right)^o$.

Remark. Recall that the counter part of this theorem for simplifying the AWGN multi-access achievability, Theorem 2.2.1, is proven using the vector interpretation of the power intervals on $\mathfrak{C}$. In that proof, we visualized the computation of the combined user as the sum of the two vectors, each representing one of the two users' specifications (as in Figure 2-4). In particular, we hold the triangle formed by the three vector fixed and placed them on $\mathfrak{C}$ corresponding to the power interval of the combined user. When we look at the location of the point at which the vectors representing the two users joint relative to $\mathfrak{C}$, we saw that the middle case corresponds to the borderline situation between the other two cases when the point is on $\mathfrak{C}$.

From the proof of the previous lemma and Figure E-3, notice that the middle case in Theorem E.2.2 also corresponds to the borderline situation between the other two cases when the point at which the vectors representing the two users joint is on $\mathfrak{C}_{\mathsf{n}}$. The remaining proof of this theorem follow identical logic as the proof of Theorem 2.2.1, and is omitted.

Now, we extend the overlapping property to sets of generalized power intervals.

Definition E.2.4. Let n denote the specification of a time-shared AWGN multi-access channel, and $\mathcal{U}$ be a set of generalized power intervals in this channel. Then $\mathcal{U}$ is said to have the *overlapping property*, or to be *overlapping*, *in this time-shared AWGN multi-access channel* if $\mathsf{T}_{\mathsf{n}}(\mathcal{V}) \subseteq \mathsf{T}_{\mathsf{n}}(\mathcal{U})$ for all $\mathcal{V} \subseteq \mathcal{U}$.

From our formulation of the time-shared AWGN multi-access achievability in (E.26), we have

Theorem E.2.3. *Let* n *denote the specification of a time-shared AWGN multi-access channel, and* $\mathcal{U}$ *be a set of generalized power intervals in this channel that has the overlapping property. Then, the (common) maximal amount that the noise variances of all the sub-channels may be raised without violating the achievability of the set of user specifications in the resulting channel is* $\eta(\mathsf{T}_{\mathsf{n}}(\mathcal{U}))$.

Recall, in Chapter 2, Theorem 2.3.7 establishes that if successively replacing two overlapping power intervals by their combined user power interval ends with a single power interval, then the given set of power intervals has the overlapping property. In particular, notice that all proofs leading to this conclusion there require no more than Theorem 2.2.1. Since we have established the counterpart of that theorem here in Theorem E.2.2 above, the same algorithm must also be available to check for the overlapping property of sets of generalized power intervals in a given time-shared AWGN multi-access channel. We formalize these results below.

Algorithm E.2.4.
Let $\mathcal{U}' = \mathcal{U}$.

1. *If there exist* T_1, $T_2 \in \mathcal{U}'$ *such that* T_1 *and* T_2 *are overlapping (i.e. either intersecting or adjacent), then let* $\mathcal{U}' = \mathcal{U}' \setminus \{T_1, T_2\} \cup \{\mathsf{T}_{\mathsf{n}}(\{T_1, T_2\})\}$, *and repeat this step*

2. *Otherwise, stop*

Theorem E.2.5. *Let* n *denote the specification of a time-shared AWGN multi-access channel, and* $\mathcal{U}$ *be a set of generalized power intervals in this channel. Then* $\mathcal{U}$ *has the overlapping property if and only if the outcome of Algorithm E.2.4 contains a single power interval.*

Therefore, realization of Algorithm E.2.4 presented in Algorithm 2.3.8 can be similarly adapted to consider sets of generalized power intervals.

Algorithm E.2.6.

Step 1: *Order members of* $\mathcal{U}$ *into* $\mathcal{U}' = \{T_i = (a_i, b_i], i \in [1, \ldots, N]\}$ *such that* $a_i \leq a_{i+1}$; *initialize* $i = 1$.

Step 2: *If T_i is the last element in $\mathcal{U}'$, exit the algorithm. Otherwise*

Step 3: *If T_i and T_{i+1} are disjoint, i.e. $b_i < a_{i+1}$, then increase i by 1 and go to Step 2.*

Step 4: *Otherwise T_i and T_{i+1} overlap. Replace the two power intervals by their combined user $\mathsf{T}_{\mathsf{n}}(\{T_i, T_{i+1}\})$. In particular, let $T_i = \mathsf{T}_{\mathsf{n}}(\{T_i, T_{i+1}\})$ and update $\mathcal{U}'$ by renaming $T_j = T_{j+1}\ \forall j \geq i+1$. If T_{i-1} exists, decrease i by 1, and go to Step 3; else go to Step 2*

As in Chapter 2, we note that this algorithm requires no more than $N - 1$ iterations to complete. Hence, its total computational requirement is bounded by the initial sort, which is $O(N \ln N)$.

Finally, we extend the irreducible equivalent construction to consider general sets of generalized power intervals in a given time-shared AWGN multi-access channel.

Definition E.2.5. Let n denote the specification of a time-shared AWGN multi-access channel, and $\mathcal{U}$ be a set of generalized power intervals in this channel. The resulting set of Algorithm E.2.4, when its members are ordered in the ascending order using Definition 2.1.3, is called the *irreducible equivalent* of $\mathcal{U}$ in this time-shared AWGN multi-access channel, and is denoted by $\mathsf{irr}_{\mathsf{n}}(\mathcal{U})$.

The i^{th} member in $\mathsf{irr}_{\mathsf{n}}(\mathcal{U})$ is denoted by $[\mathsf{irr}_{\mathsf{n}}(\mathcal{U})]_i$.

Definition E.2.6. The set of generalized power intervals in $\mathcal{U}$ that are contained in $[\mathsf{irr}_{\mathsf{n}}(\mathcal{U})]_i$ is termed the *the i^{th} constructing set* of $\mathsf{irr}_{\mathsf{n}}(\mathcal{U})$, and denoted by $\mathsf{B}_{\mathsf{n},i}(\mathcal{U})$.

Corollary E.2.7. *Let n denote the specification of a time-shared AWGN multi-access channel, and $\mathcal{U}$ be a set of generalized power intervals in this channel. Then, the (common) maximal amount that the noise variances of all the sub-channels may be raised without violating the achievability of the set of user specifications in the resulting channel is the lower boundary of* $[\mathsf{irr}_{\mathsf{n}}(\mathcal{U})]_1$.

Since Algorithm E.2.4 require no more than $O(N \ln N)$ computations to complete, we conclude that checking the achievability of a set of N user specifications in a time-shared AWGN multi-access channel requires no more than $O(N \ln N)$ computations. This completes our simplification of the time-shared AWGN multi-access achievability.

Appendix F

The Time portions of the Achieving sets Resulting from the Second Set of Operations may Not be Ordered

Theorem F.0.8. *Let a set of two time-power bi-intervals* $\widehat{\mathcal{V}} = \{(a_1, a_1 + p] \times (0, \alpha], (a_2, a_2 + p] \times (\alpha, 1]\}$ *be an achieving set of* $(a, a+p]$ *with* $a_1 \geq a \geq a_2 > 0$ *and* $\alpha \in [0,1]$. *Then for any* $\delta \in [0, p]$, *there is*

$$\bar{\mathsf{r}}\left(\widehat{\mathsf{G}}\left(\widehat{\mathcal{V}}, \delta\right)\right) \geq \mathsf{r}\left(\mathsf{G}\left((a, a+p], \delta\right)\right) \tag{F.1}$$

$$\bar{\mathsf{r}}\left(\widehat{\mathsf{L}}\left(\widehat{\mathcal{V}}, \delta\right)\right) \leq \mathsf{r}\left(\mathsf{L}\left((a, a+p], \delta\right)\right) \tag{F.2}$$

First, we prove two lemmas.

Lemma F.0.9. *For any given* $a > 0$, $0 < d \leq a$, *and* $\delta \geq 0$,

$$\frac{1/(a+\delta)}{1/a} \geq \frac{\ln\left(\frac{a+\delta}{a+\delta-d}\right)}{\ln\left(\frac{a}{a-d}\right)} \geq \frac{1/(a+\delta-d)}{1/(a-d)} \tag{F.3}$$

Discussion. Sanity check:

$$
\begin{aligned}
\frac{1/(a+\delta)}{1/a} - \frac{1/(a+\delta-d)}{1/(a-d)} &= \frac{a}{a+\delta} - \frac{a-d}{a+\delta-d} \\
&= -\frac{\delta}{a+\delta} + \frac{\delta}{a+\delta-d} \\
&= \frac{\delta d}{(a+\delta)(a+\delta-d)} \geq 0
\end{aligned}
$$

Proof. To show the first inequality, first cross multiply the terms and move the result to the same side to obtain the following equivalent inequality.

$$
\Lambda = a \ln\left(\frac{a}{a-d}\right) - (a+\delta)\ln\left(\frac{a+\delta}{a+\delta-d}\right) \geq 0 \tag{F.4}
$$

I will now show $\min \Lambda \geq 0$ over all possible a, d and δ.

Consider the partial differential of Λ with respect to δ.

$$
\begin{aligned}
\frac{\partial \Lambda}{\partial \delta} &= -\ln\left(\frac{a+\delta}{a+\delta-d}\right) - (a+\delta)\left(\frac{1}{a+\delta} - \frac{1}{a+\delta-d}\right) \\
&= -\ln\left(\frac{a+\delta}{a+\delta-d}\right) + \frac{d}{a+\delta-d} \qquad \text{(F.5)}\\
\frac{\partial^2 \Lambda}{\partial \delta^2} &= -\left(\frac{1}{a+\delta} - \frac{1}{a+\delta-d}\right) - \frac{d}{(a+\delta-d)^2} \\
&= \frac{d}{(a+\delta)(a+\delta-d)} - \frac{d}{(a+\delta-d)^2} \\
&= \frac{d}{a+\delta-d}\left(\frac{1}{a+\delta} - \frac{1}{a+\delta-d}\right) \\
&= \frac{d}{(a+\delta-d)} \frac{-d}{(a+\delta)(a+\delta-d)} \leq 0 \qquad \text{(F.6)}
\end{aligned}
$$

Therefore, the minimum of $\frac{\partial \Lambda}{\partial \delta}$ for any set of give a and d occurs as δ goes to infinity. Now, since

$$
\min \frac{\partial \Lambda}{\partial \delta} = \lim_{\delta \to \infty} \frac{\partial \Lambda}{\partial \delta} = \lim_{\delta \to \infty} -\ln\left(\frac{a+\delta}{a+\delta-d}\right) + \frac{d}{a+\delta-d} = 0 \tag{F.7}
$$

we conclude that $\frac{\partial \Lambda}{\partial \delta} \geq 0$. In other words, the minimum of Λ for given a and d occurs at $\delta = 0$, i.e.

$$
\min \Lambda = \lim_{\delta \to 0} \Lambda = \lim_{\delta \to 0} a \ln\left(\frac{a}{a-d}\right) - (a+\delta)\ln\left(\frac{a+\delta}{a+\delta-d}\right) = 0 \tag{F.8}
$$

Thus completing the proof of the first inequality.

To show the second inequality, cross multiply the terms and move the result to the same side to obtain the following equivalent inequality.

$$\Gamma = (a+\delta-d)\ln\left(\frac{a+\delta}{a+\delta-d}\right) - (a-d)\ln\left(\frac{a}{a-d}\right) \geq 0 \tag{F.9}$$

I will now show that $\min\Gamma \geq 0$ over all possible a, d and δ.

Consider the partial differential of Γ with respect to δ.

$$\begin{aligned}\frac{\partial\Gamma}{\partial\delta} &= \ln\left(\frac{a+\delta}{a+\delta-d}\right) + (a+\delta-d)\left(\frac{1}{a+\delta} - \frac{1}{a+\delta-d}\right)\\ &= \ln\left(\frac{a+\delta}{a+\delta-d}\right) - \frac{d}{a+\delta} \qquad \text{(F.10)}\\ \frac{\partial^2\Gamma}{\partial\delta^2} &= \frac{1}{a+\delta} - \frac{1}{a+\delta-d} + \frac{d}{(a+\delta)^2}\\ &= -\frac{d}{(a+\delta)(a+\delta-d)} - \frac{d}{(a+\delta)^2}\\ &= \frac{d}{a+\delta}\left(\frac{1}{a+\delta} - \frac{1}{a+\delta-d}\right)\\ &= \frac{d}{(a+\delta)}\,\frac{-d}{(a+\delta)(a+\delta-d)} \leq 0 \qquad \text{(F.11)}\end{aligned}$$

Therefore, the minimum of $\frac{\partial\Gamma}{\partial\delta}$ for any pair of given a and d occurs as δ goes to infinity. Now, since

$$\min\frac{\partial\Gamma}{\partial\delta} = \lim_{\delta\to\infty}\frac{\partial\Gamma}{\partial\delta} = \lim_{\delta\to\infty}\ln\left(\frac{a+\delta}{a+\delta-d}\right) - \frac{d}{a+\delta} = 0 \tag{F.12}$$

we conclude that $\frac{\partial\Gamma}{\partial\delta} \geq 0$. In other words, the minimum of λ for given a and d occurs at $\delta = 0$, i.e.

$$\min\Gamma = \lim_{\delta\to 0}\Gamma = \lim_{\delta\to 0}(a+\delta-d)\ln\left(\frac{a+\delta}{a+\delta-d}\right) - (a-d)\ln\left(\frac{a}{a-d}\right) = 0 \tag{F.13}$$

Thus completing the proof of this lemma. □

Lemma F.0.10. *For any given* $a_1 \geq a \geq a_2 > 0$, *and* δ, $p \geq 0$, *there is*

$$\frac{\ln\left(\frac{a}{a_2}\right)}{\ln\left(\frac{a_1}{a}\right)} \geq \frac{\ln\left(\frac{a+\delta}{a_2+\delta}\right)}{\ln\left(\frac{a_1+\delta}{a+\delta}\right)} \tag{F.14}$$

Proof. Let $d_2 \geq 0$ and $0 \leq d_1 < a$ be such that

$$a_1 = a + d_2 \tag{F.15}$$

$$a_2 = a - d_1 \tag{F.16}$$

Hence (F.14) is equivalent to

$$\frac{\ln\left(\frac{a}{a-d_1}\right)}{\ln\left(\frac{a+d_2}{a}\right)} \geq \frac{\ln\left(\frac{a+\delta}{a+\delta-d_1}\right)}{\ln\left(\frac{a+\delta+d_2}{a+\delta}\right)} \tag{F.17}$$

Exchanging the denominator of the left side with the numerator of the right side to obtain

$$\frac{\ln\left(\frac{a}{a-d_1}\right)}{\ln\left(\frac{a+\delta}{a+\delta-d_1}\right)} \geq \frac{\ln\left(\frac{a+d_2}{a}\right)}{\ln\left(\frac{a+\delta+d_2}{a+\delta}\right)} \tag{F.18}$$

Now, define the following

$$L_1 = \frac{\ln\left(\frac{a}{a-d_1}\right)}{\ln\left(\frac{a+\delta}{a+\delta-d_1}\right)} \qquad L_2 = \frac{\ln\left(\frac{a+d_2}{a}\right)}{\ln\left(\frac{a+\delta+d_2}{a+\delta}\right)} \tag{F.19}$$

So we need to prove $L_1 \geq L_2$.

Consider their partial derivatives with respect to d_1 and d_2 respectively.

$$\frac{\partial L_1}{\partial d_1} = \left(\frac{1}{a-d_1}\ln\left(\frac{a+\delta}{a+\delta-d_1}\right) - \frac{1}{a+\delta-d_1}\ln\left(\frac{a}{a-d_1}\right)\right) / \left(\ln\left(\frac{a+\delta}{a+\delta-d_1}\right)\right)^2 \tag{F.20}$$

$$\frac{\partial L_2}{\partial d_2} = \left(\frac{1}{a+d_2}\ln\left(\frac{a+\delta+d_2}{a+\delta}\right) - \frac{1}{a+\delta+d_2}\ln\left(\frac{a+d_2}{a}\right)\right) / \left(\ln\left(\frac{a+\delta+d_2}{a+\delta}\right)\right)^2 \tag{F.21}$$

(F.22)

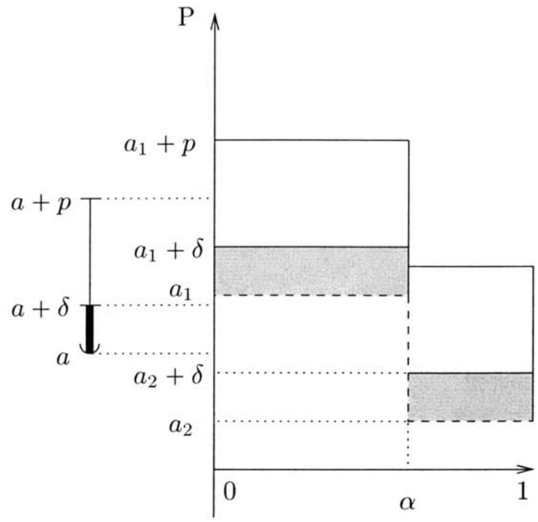
By the first inequality of Lemma F.0.9, $\frac{\partial L_2}{\partial d_2} \leq 0$. Thus, for fixed a and δ,

$$\min L_1 = \lim_{d_1 \to 0} L_1 = \lim_{d_1 \to 0} \frac{1/(a-d_1)}{1/(a+\delta-d_1)} = \frac{a+\delta}{a} \tag{F.23}$$

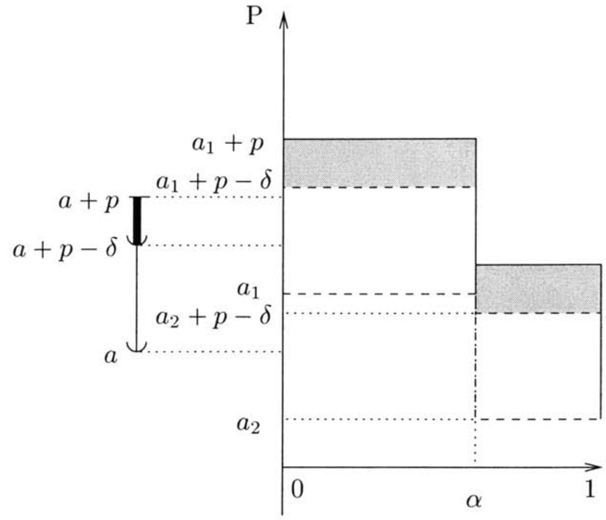
By the second inequality of Lemma F.0.9, $\frac{\partial L_1}{\partial d_1} \geq 0$. Thus, for fixed a and δ,

$$\max L_2 = \lim_{d_2 \to 0} L_2 = \lim_{d_2 \to 0} \frac{1/(a+d_2)}{1/(a+\delta+d_2)} = \frac{a+\delta}{a} \tag{F.24}$$

Finally, since $\min L_1 = \frac{a+\delta}{a} = \max L_2$, we have $L_1 \geq L_2$. □



(a) First Inequality



(b) Second Inequality

Figure F-1: The Time-Power Intervals in Theorem F.0.8

Proof of Theorem F.0.8: Figure F-1 illustrates the time-power bi-intervals concerned in this theorem.

Observe that

$$\widehat{\mathsf{G}}(\mathcal{V},\delta) = \{(a_1, a_1+\delta] \times (0,\alpha], (a_2, a_2+\delta \times (\alpha, 1]\} \tag{F.25}$$

$$\mathsf{G}\left((a, a+p], \delta\right) = (a, a+\delta] \tag{F.26}$$

Thus (F.1) is equivalent to

$$\frac{\alpha}{2}\ln\left(\frac{a_1+\delta}{a_1}\right) + \frac{(1-\alpha)}{2}\ln\left(\frac{a_2+\delta}{a_2}\right) \geq \frac{1}{2}\ln\left(\frac{a+\delta}{a}\right) \tag{F.27}$$

If $a_2 = a$, observe that the assumption that $\widehat{\mathcal{V}}$ is an achieving set of $(a, a+p]$ dictates that $a_1 = a$. In such cases, the result of this theorem is trivially true. So in the rest of the proof, we assume $a_1 > a > a_2 > 0$.

Define the following function

$$\mathsf{A}(\delta) = \frac{\ln\left(\frac{a}{a+\delta}\,\frac{a_2+\delta}{a_2}\right)}{\ln\left(\frac{a_1}{a_1+\delta}\,\frac{a_2+\delta}{a_2}\right)} \tag{F.28}$$

By assumption, $\widehat{\mathcal{V}}$ is an achieving set of $(a, a+p]$. Hence, there is

$$\frac{\alpha}{2}\ln\left(\frac{a_1+p}{a_1}\right) + \frac{(1-\alpha)}{2}\ln\left(\frac{a_2+p}{a_2}\right) = \frac{1}{2}\ln\left(\frac{a+p}{a}\right) \tag{F.29}$$

After some algebraic manipulations, we have

$$\alpha = \frac{\ln\left(\frac{a+p}{a}\frac{a_2}{a_2+p}\right)}{\ln\left(\frac{a_1+p}{a_1}\frac{a_2}{a_2+p}\right)} = \frac{\ln\left(\frac{a}{a+p}\frac{a_2+p}{a_2}\right)}{\ln\left(\frac{a_1}{a_1+p}\frac{a_2+p}{a_2}\right)} \tag{F.30}$$

The second equality is so that both the numerator and the denominator are positive (using the monotonicity of $\mathsf{r}(\cdot)$). In other words, $\mathsf{A}(p) = \alpha$.

What remains to be done is to show that $\mathsf{A}(x)$ is a decreasing function of x. Once this is accomplished, there is $\mathsf{A}(\delta) \geq \alpha$. Hence

$$\frac{\ln\left(\frac{a}{a+\delta}\frac{a_2+\delta}{a_2}\right)}{\ln\left(\frac{a_1}{a_1+\delta}\frac{a_2+\delta}{a_2}\right)} \geq \alpha \tag{F.31}$$

And (F.27) follows.

Now, I will show that $\mathsf{A}(\delta)$ is a decreasing function of δ, for $\delta \in (0, p]$.

From the definition of $\mathsf{A}(\delta)$, there is:

$$\mathsf{A}(\delta)\ln\left(\frac{a_1}{a_2}\right) - \ln\left(\frac{a}{a_2}\right) = \mathsf{A}(\delta)\ln\left(\frac{a_1+\delta}{a_2+\delta}\right) - \ln\left(\frac{a+\delta}{a_2+\delta}\right) \tag{F.32}$$

Define the following function:

$$f(\alpha, \delta) = \alpha\ln\left(\frac{a_1+\delta}{a_2+\delta}\right) - \ln\left(\frac{a+\delta}{a_2+\delta}\right) \tag{F.33}$$

Let $\mathcal{L}(\delta)$ denote the segment of the line $(\alpha, f(\alpha,\delta))$ for $\alpha \in [0,1]$, i.e.

$$\mathcal{L}(\delta) = \{(\alpha, f(\alpha,\delta)), \alpha \in [0,1]\}$$

With these definitions, $\mathsf{A}(\delta)$ may be determined as the horizontal dimension of the intersection of two straight line segments $\mathcal{L}(0)$ and $\mathcal{L}(\delta)$, as in Figure F-2.

Notice that the line $\mathcal{L}(\delta)$ intersects the α axis at $\frac{\ln\left(\frac{a+\delta}{a_2+\delta}\right)}{\ln\left(\frac{a_1+\delta}{a_2+\delta}\right)}$.

By Lemma F.0.10, $\frac{\ln\left(\frac{a+\delta}{a_2+\delta}\right)}{\ln\left(\frac{a_1+\delta}{a+\delta}\right)}$ is a decreasing function in δ for fixed $a_1 \geq a \geq a_2 \geq 0$.

Since for positive function $g(x)$, $\frac{dg(x)}{dx} \leq 0$ implies $\frac{d}{dx}\frac{g(x)}{1+g(x)} \leq 0$, we conclude that the

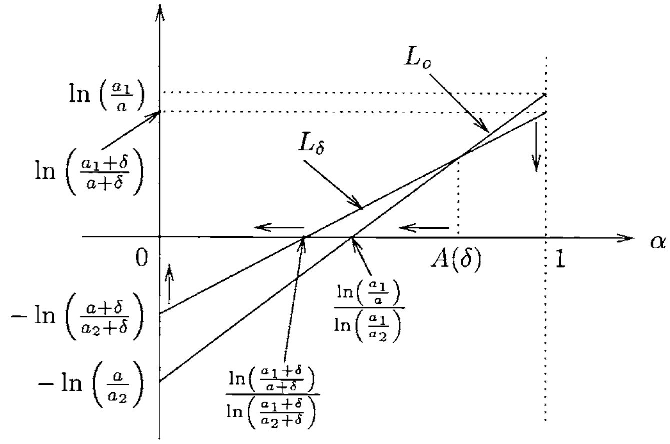


Figure F-2: Determining $\mathsf{A}(\delta)$ Graphically

intersect of the line $\mathcal{L}(\delta)$ with the α axis, $\frac{\ln\left(\frac{a+\delta}{a_2+\delta}\right)}{\ln\left(\frac{a_1+\delta}{a_2+\delta}\right)}$, is a decreasing function of δ.

Finally, since the magnitude of both ends of $\mathcal{L}(\delta)$, $\frac{a+\delta}{a_2+\delta}$ and $\frac{a_1+\delta}{a+\delta}$, are decreasing functions of δ (see the arrows on Figure F-2), we conclude that $\mathsf{A}(\delta)$ is a decreasing function of δ, and this establishes (F.1).

To show (F.2), observe that

$$\widehat{\mathsf{L}}(\widehat{\mathcal{V}},\delta) = \widehat{\mathsf{cmpl}}(\widehat{\mathcal{V}},\widehat{\mathsf{G}}(\widehat{\mathcal{V}},p-\delta)) \tag{F.34}$$

$$\mathsf{L}\left((a,a+p],\delta\right) = \mathsf{cmpl}((a,a+p],\mathsf{G}\left((a,a+p],\delta\right)) \tag{F.35}$$

By assumption,

$$\mathsf{r}((a,a+p]) = \bar{\mathsf{r}}(\widehat{\mathcal{V}}) \tag{F.36}$$

Therefore, we have

$$\begin{aligned}
\bar{\mathsf{r}}(\widehat{\mathsf{L}}(\widehat{\mathcal{V}},\delta)) &= \bar{\mathsf{r}}(\widehat{\mathcal{V}}) - \bar{\mathsf{r}}(\widehat{\mathsf{G}}(\widehat{\mathcal{V}},,p-\delta)) \\
&\le \bar{\mathsf{r}}(\widehat{\mathcal{V}}) - \mathsf{r}(\mathsf{G}\left((a,a+p],p-\delta\right)) \\
&= \mathsf{r}((a,a+p]) - \mathsf{r}(\mathsf{G}\left((a,a+p],p-\delta\right)) \\
&= \mathsf{r}(\mathsf{L}\left((a,a+p],\delta\right))
\end{aligned} \tag{F.37}$$

where the inequality at the second step is from (F.1). This completes the proof of this theorem.

□

Incidentally, notice that

$$\lim_{\delta\to\infty} \mathsf{A}(\delta) = \frac{\ln(a_1/a)}{\ln(a_1/a_2)} \tag{F.38}$$

Corollary F.0.11. *Let a power interval $(a,b]$ be simply achieved by a set disjoint time-power intervals $\widehat{\mathcal{V}}$. Then for any $\delta \in [0, b-a]$, there is*

$$\bar{\mathsf{r}}\left(\widehat{\mathsf{L}}\left(\widehat{\mathcal{V}},\delta\right)\right) \leq \mathsf{r}(\mathsf{L}((a,b],\delta)) \leq \mathsf{r}(\mathsf{G}((a,b],\delta)) \leq \bar{\mathsf{r}}\left(\widehat{\mathsf{G}}\left(\widehat{\mathcal{V}},\delta\right)\right) \tag{F.39}$$

Proof. Let $\widehat{\mathcal{V}}$ be a set of K disjoint time-power intervals. For each $k \in [1..K]$, construct $\widehat{S}_k = \widehat{\mathsf{T}}(\mathsf{head}(\widehat{\mathcal{V}},k))$, and $A_k = \widehat{\mathsf{I}}_t(\widehat{S}_k)$. Observe that $\widehat{S}_k$ and $\widehat{T}_l$ have disjoint time intervals for all $l > k$.

Observe that for $k \in [2..K]$, take $\widehat{\mathcal{V}}'_k = \{\widehat{S}_{k-1}, \widehat{T}_k\}$. Observe that $\widehat{S}_k = \widehat{\mathsf{T}}(\mathcal{W}_k)$. By Theorem F.0.8, there is:

$$\bar{\mathsf{r}}\left(\widehat{\mathsf{L}}\left(\widehat{\mathcal{V}}'_k,\delta\right)\right) \leq \bar{\mathsf{r}}\left(\widehat{\mathsf{L}}\left(\widehat{S}_k,\delta\right)\right) \leq \bar{\mathsf{r}}\left(\widehat{\mathsf{G}}\left(\widehat{S}_k,\delta\right)\right) \leq \bar{\mathsf{r}}\left(\widehat{\mathsf{G}}\widehat{\mathcal{V}}'_k\delta\right) \tag{F.40}$$

Moreover, since $\widehat{S}_{k-1}$ and $\widehat{T}_k$, there is

$$\bar{\mathsf{r}}\left(\widehat{\mathsf{L}}\left(\widehat{\mathcal{V}}'_k,\delta\right)\right) = \bar{\mathsf{r}}\left(\widehat{\mathsf{L}}\left(\widehat{S}_{k-1},\delta\right)\right) + \bar{\mathsf{r}}\left(\widehat{\mathsf{L}}\left(\widehat{T}_i,\delta\right)\right) \tag{F.41}$$

$$\bar{\mathsf{r}}\left(\widehat{\mathsf{G}}\left(\widehat{\mathcal{V}}'_k,\delta\right)\right) = \bar{\mathsf{r}}\left(\widehat{\mathsf{G}}\left(\widehat{S}_{k-1},\delta\right)\right) + \bar{\mathsf{r}}\left(\widehat{\mathsf{G}}\left(\widehat{T}_i,\delta\right)\right) \tag{F.42}$$

Successively substitute (F.41) and (F.42) into (F.40), we obtain

$$\bar{\mathsf{r}}\left(\widehat{\mathsf{L}}\left(\widehat{\mathcal{V}},\delta\right)\right) \leq \mathsf{r}(\mathsf{L}((a,b],\delta)) \leq \mathsf{r}(\mathsf{G}((a,b],\delta)) \leq \bar{\mathsf{r}}\left(\widehat{\mathsf{G}}\left(\widehat{\mathcal{V}},\delta\right)\right) \tag{F.43}$$

This completes the proof. □

Appendix G

Proof of Theorem 5.2.5 and Theorem 5.2.6

First, we prove Theorem 5.2.5. For readers' convenience, the theorem is copied below.

Theorem 5.2.5. *Let $\mathcal{V} = \{T_i, i \in [1..m]\}$ be a set of disjoint power intervals with $T_i < T_{i+1}$. Let $\widehat{\mathsf{W}}$ be an achieving set of $\mathsf{T}(\mathcal{V})$. Suppose*

$$\sum_{j=1}^{i} \mathsf{r}(T_j) \leq \bar{\mathsf{r}}\left(\widehat{\mathsf{G}}\left(\widehat{\mathsf{W}}, \sum_{j=1}^{i} \mathsf{p}(T_j)\right)\right), \quad \forall i \in [1..m] \tag{G.1}$$

Then for all $\delta \in [0, \mathsf{p}(\mathsf{T}(\mathcal{V}))]$,

$$\mathsf{r}(\mathsf{G}(\mathcal{V}, \delta)) \leq \bar{\mathsf{r}}(\widehat{\mathsf{G}}(\widehat{\mathsf{W}}, \delta)) \tag{G.2}$$

Lemma G.0.12. *Let $(a, a+p]$ be the combined user of $(a_1, a_1 + p_1]$ and $(a_2, a_2 + p_2]$. Let $(a'_1, a'_1 + p_1]$ and $(a'_2, a'_2 + p_2]$ also have combined user $(a, a+p]$. Suppose $a'_1 \leq a_1$. Then for any $\delta \in [0, p_2]$, there is*

$$\mathsf{r}((a_1, a_1 + p_1]) + \mathsf{r}((a_2, a_2 + \delta]) \leq \mathsf{r}((a'_1, a'_1 + p_1]) + \mathsf{r}((a'_2, a'_2 + \delta]) \tag{G.3}$$

Proof. This lemma is equivalent to showing that

$$\mathsf{r}((a_1, a_1 + p_1]) + \mathsf{r}((a_2, a_2 + \delta]) \tag{G.4}$$

subject to

$$\mathsf{T}\left(\{(a_1, a_1 + p_1], (a_2, a_2 + p_2]\}\right) = (a, a + p] \tag{G.5}$$

is a decreasing function with the increase of a_1.

From (G.5), there is

$$\frac{a_1 + p_1}{a_1} \frac{a_2 + p_2}{a_2} = \frac{a + p}{a} \tag{G.6}$$

Let $c = \frac{a+p}{a}$, there is

$$a_2 = \frac{p_2}{\frac{ca_1}{a_1+p_1} - 1} \tag{G.7}$$

By definition of r() function, there is the following for (G.4):

$$\mathsf{r}((a_1, a_1 + p_1]) + \mathsf{r}((a_2, a_2 + \delta]) = \frac{1}{2} \ln\left(\frac{a_1 + p_1}{a_1} \frac{a_2 + \delta}{a_2}\right) \tag{G.8}$$

Hence, we may equivalently show that the product

$$\frac{a_1 + p_1}{a_1} \frac{a_2 + \delta}{a_2} \tag{G.9}$$

decreases with the increase of a_1 subject to condition (G.7).

Substitute (G.7) into (G.9), there is

$$\begin{aligned} \frac{a_1 + p_1}{a_1} \frac{a_2 + \delta}{a_2} &= \frac{a_1 + p_1}{a_1} \frac{\frac{p_2}{\frac{ca_1}{a_1+p_1} - 1} + \delta}{\frac{p_2}{\frac{ca_1}{a_1+p_1} - 1}} \\ &= \frac{a_1 + p_1}{a_1} \frac{p_2 + \delta\left(\frac{ca_1}{a_1+p_1} - 1\right)}{p_2} \\ &= \frac{1}{p_2}\left((p_2 - \delta)\frac{a_1 + p_1}{a_1} + \delta c\right) \end{aligned} \tag{G.10}$$

Observe that $\frac{a_1+p_1}{a_1} \frac{a_2+\delta}{a_2} \geq 0$ for all $\delta \in [0, p_2]$.

Taking the partial differential of $\frac{a_1+p_1}{a_1}\frac{a_2+\delta}{a_2}$ with respect to a_1.

$$\begin{aligned}\frac{\partial}{\partial a_1}\left(\frac{a_1+p_1}{a_1}\frac{a_2+\delta}{a_2}\right) &= \frac{p_2-\delta}{p_2}\left(\frac{1}{a_1}-\frac{a_1+p_1}{a_1^2}\right)\\ &= \frac{p_2-\delta}{p_2}\left(-\frac{p_1}{a_1^2}\right)\\ &= -\frac{p_1(p_2-\delta)}{a_1^2 p_1}\end{aligned} \tag{G.11}$$

Since $\delta \leq p_2$, we have

$$\frac{\partial}{\partial a_1}\left(\frac{a_1+p_1}{a_1}\frac{a_2+\delta}{a_2}\right) \leq 0 \tag{G.12}$$

This completes the proof. □

Lemma G.0.13. *For a given power interval T, let $\widehat{\mathbb{W}}$ be a simple set of time-power bi-intervals with power specification $\mathsf{p}(T)$. Suppose further that*

$$\mathsf{r}(T) \leq \bar{\mathsf{r}}(\widehat{\mathbb{W}}) \tag{G.13}$$

Then for all $\delta \in [0, \mathsf{p}(T)]$

$$\mathsf{r}(\mathsf{G}(T,\delta)) \leq \bar{\mathsf{r}}(\widehat{\mathsf{G}}(\widehat{\mathbb{W}},\delta)) \tag{G.14}$$

Proof. Without the loss of generality, take

$$\widehat{\mathbb{W}} = \{(a_i, a_i + \mathsf{p}(T)] \times (\alpha_{i-1}, \alpha_i], i \in [1..m]\} \tag{G.15}$$

with $0 = \alpha_0 \leq \alpha_1 \leq \ldots \leq \alpha_m = 1$.

Define

$$\widehat{\mathbb{W}}(x) = \{(a_i + x, a_i + x + \mathsf{p}(T)] \times (\alpha_{i-1}, \alpha_i], i \in [1..m]\} \tag{G.16}$$

Notice that

$$\bar{\mathsf{r}}(\widehat{\mathbb{W}}(0)) = \sum_{i=1}^{m} \frac{\alpha_i - \alpha_{i-1}}{2} \ln\left(\frac{a_i + \mathsf{p}(T)}{a_i}\right) = \bar{\mathsf{r}}(\widehat{\mathbb{W}}) \tag{G.17}$$

Therefore, by assumption,

$$\mathsf{r}(T) \le \bar{\mathsf{r}}(\widehat{\mathsf{W}}(0)) \tag{G.18}$$

Observe that $\bar{\mathsf{r}}(\widehat{\mathsf{W}}(x))$ decreases continuously to 0 as x increases to infinity. Therefore, there exists an $x^* \ge 0$ such that

$$\mathsf{r}(T) = \bar{\mathsf{r}}(\widehat{\mathsf{W}}(x^*)) \tag{G.19}$$

In other words, T is achieved by $\widehat{\mathsf{W}}(x^*)$ simply.

By Corollary F.0.11, for all $\delta \in [0, \mathsf{p}(T)]$,

$$\mathsf{r}(\mathsf{G}(T, \delta)) \le \bar{\mathsf{r}}(\widehat{\mathsf{G}}(\widehat{\mathsf{W}}(x^*), \delta)) \tag{G.20}$$

There is also

$$\begin{aligned} \bar{\mathsf{r}}(\widehat{\mathsf{G}}(\widehat{\mathsf{W}}(x^*), \delta)) &= \sum_{i=1}^{m} \frac{\alpha_i - \alpha_{i-1}}{2} \ln\left(\frac{a_i + x^* + \delta}{a_i + x^*}\right) \\ &\le \sum_{i=1}^{m} \frac{\alpha_i - \alpha_{i-1}}{2} \ln\left(\frac{a_i + \delta}{a_i}\right) \\ &= \bar{\mathsf{r}}(\widehat{\mathsf{G}}(\widehat{\mathsf{W}}, \delta)) \end{aligned} \tag{G.21}$$

Combining (G.20) and (G.21), we have the desired inequality for this lemma. □

Lemma G.0.14. *Let T_1 and T_2 be two disjoint power intervals with $T_1 < T_2$. Let $\widehat{\mathsf{W}}$ be a set of time-power bi-intervals such that*

$$\mathsf{r}(T_1) \le \bar{\mathsf{r}}(\widehat{\mathsf{G}}(\widehat{\mathsf{W}}, \mathsf{p}(T_1))) \tag{G.22}$$

$$\mathsf{r}(T_1) + \mathsf{r}(T_2) \le \bar{\mathsf{r}}(\widehat{\mathsf{G}}(\widehat{\mathsf{W}}, \mathsf{p}(T_1) + \mathsf{p}(T_2))) \tag{G.23}$$

Then for any $\delta \in [0, \mathsf{p}(T_2)]$,

$$\mathsf{r}(T_1) + \mathsf{r}(\mathsf{G}(T_2, \delta)) \le \bar{\mathsf{r}}(\widehat{\mathsf{G}}(\widehat{\mathsf{W}}, \mathsf{p}(T_1) + \delta)) \tag{G.24}$$

Proof. If $\mathsf{r}(T_1) + \mathsf{r}(T_2) \le \bar{\mathsf{r}}(\widehat{\mathsf{G}}(\widehat{\mathsf{W}}, \mathsf{p}(T_1)))$, the lemma is trivially true.

Otherwise, let T_1^* and T_2^* be such that

$$\mathsf{p}(T_1^*) = \mathsf{p}(T_1) \tag{G.25}$$

$$\mathsf{p}(T_2^*) = \mathsf{p}(T_2) \tag{G.26}$$

$$\mathsf{r}(T_1^*) = \bar{\mathsf{r}}(\widehat{\mathsf{G}}(\widehat{\mathcal{W}}, \mathsf{p}(T_1))) \tag{G.27}$$

$$\mathsf{r}(T_1^*) + \mathsf{r}(T_2^*) = \mathsf{r}(T_1) + \mathsf{r}(T_2) \tag{G.28}$$

By Lemma G.0.12, we have for all $\delta \in [0, \mathsf{p}(T_2)]$

$$\mathsf{r}(T_1) + \mathsf{r}(\mathsf{G}(T_2, \delta)) \leq \mathsf{r}(T_1^*) + \mathsf{r}(\mathsf{G}(T_2^*, \delta)) \tag{G.29}$$

Thus the lemma is proven if we can show

$$\begin{aligned} \mathsf{r}(T_1^*) + \mathsf{r}(\mathsf{G}(T_2^*, \delta)) &\leq \bar{\mathsf{r}}(\widehat{\mathsf{G}}(\widehat{\mathcal{W}}, \mathsf{p}(T_1) + \delta)) \\ &= \bar{\mathsf{r}}(\widehat{\mathsf{G}}(\widehat{\mathcal{W}}, \mathsf{p}(T_1^*) + \delta)) \end{aligned} \tag{G.30}$$

Combining (G.23) and (G.28),

$$\mathsf{r}(T_1^*) + \mathsf{r}(T_2^*) \leq \bar{\mathsf{r}}(\widehat{\mathsf{G}}(\widehat{\mathcal{W}}, \mathsf{p}(T_1) + \mathsf{p}(T_2))) \tag{G.31}$$

Therefore

$$\mathsf{r}(T_2^*) \leq \mathsf{r}(\widehat{\mathsf{G}}(\widehat{\mathcal{W}}, \mathsf{p}(T_1^*) + \mathsf{p}(T_2^*))) - \mathsf{r}(T_1^*) \tag{G.32}$$

Substitute (G.27) into the above inequality to get

$$\begin{aligned} \mathsf{r}(T_2^*) &\leq \bar{\mathsf{r}}(\widehat{\mathsf{G}}(\widehat{\mathcal{W}}, \mathsf{p}(T_1^*) + \mathsf{p}(T_2^*))) - \bar{\mathsf{r}}(\widehat{\mathsf{G}}(\widehat{\mathcal{W}}, \mathsf{p}(T_1^*))) \\ &= \bar{\mathsf{r}}(\widehat{\mathsf{cmpl}}(\widehat{\mathsf{G}}(\widehat{\mathcal{W}}, \mathsf{p}(T_1^*) + \mathsf{p}(T_2^*)), \widehat{\mathsf{G}}(\widehat{\mathcal{W}}, \mathsf{p}(T_1^*)))) \\ &= \bar{\mathsf{r}}(\widehat{\mathsf{G}}(\widehat{\mathsf{cmpl}}(\widehat{\mathcal{W}}, \widehat{\mathsf{G}}(\widehat{\mathcal{W}}, \mathsf{p}(T_1^*))), \mathsf{p}(T_2^*))) \end{aligned} \tag{G.33}$$

By Lemma G.0.13, for all $\delta \in [0, \mathsf{p}(T_2)]$,

$$\begin{aligned} \mathsf{r}(\mathsf{G}(T_2^*), \delta) &\leq \bar{\mathsf{r}}(\widehat{\mathsf{G}}(\widehat{\mathsf{G}}(\widehat{\mathsf{cmpl}}(\widehat{\mathcal{W}}, \widehat{\mathsf{G}}(\widehat{\mathcal{W}}, \mathsf{p}(T_1^*))), \mathsf{p}(T_2^*)), \delta)) \\ &= \bar{\mathsf{r}}(\widehat{\mathsf{G}}(\widehat{\mathsf{cmpl}}(\widehat{\mathcal{W}}, \widehat{\mathsf{G}}(\widehat{\mathcal{W}}, \mathsf{p}(T_1^*))), \delta)) \end{aligned} \tag{G.34}$$

Add (G.27) and (G.34),

$$\begin{aligned} \mathsf{r}(T_1^*) + \mathsf{r}(\mathsf{G}(T_2^*), \delta) &\leq \bar{\mathsf{r}}(\widehat{\mathsf{G}}(\widehat{\mathcal{W}}, \mathsf{p}(T_1))) + \bar{\mathsf{r}}(\widehat{\mathsf{G}}(\widehat{\mathsf{cmpl}}(\widehat{\mathcal{W}}, \widehat{\mathsf{G}}(\widehat{\mathcal{W}}, \mathsf{p}(T_1^*))), \delta)) \\ &= \bar{\mathsf{r}}(\widehat{\mathsf{G}}(\widehat{\mathcal{W}}, \mathsf{p}(T_1^*))) + \bar{\mathsf{r}}(\widehat{\mathsf{G}}(\widehat{\mathsf{cmpl}}(\widehat{\mathcal{W}}, \widehat{\mathsf{G}}(\widehat{\mathcal{W}}, \mathsf{p}(T_1^*))), \delta)) \\ &= \bar{\mathsf{r}}(\widehat{\mathsf{G}}(\widehat{\mathcal{W}}, \mathsf{p}(T_1^*) + \delta)) \end{aligned} \tag{G.35}$$

This completes the proof. □

Proof of Theorem 5.2.5: Let $m^* \in [1..m]$ be such that

$$\sum_{i=1}^{m^*} \mathsf{p}(T_i) \leq \delta \leq \sum_{i=1}^{m^*+1} \mathsf{p}(T_i) \tag{G.36}$$

Take $S_{m^*} = \mathsf{T}(\mathsf{head}(\mathcal{V}, m^*))$. By assumption, there are

$$\mathsf{r}(S_{m^*}) = \sum_{j=1}^{m^*} \mathsf{r}(T_j) \leq \bar{\mathsf{r}}\left(\widehat{\mathsf{G}}\left(\widehat{\mathcal{W}}, \sum_{j=1}^{m^*} \mathsf{p}(T_j)\right)\right) = \bar{\mathsf{r}}\left(\widehat{\mathsf{G}}\left(\widehat{\mathcal{W}}, \mathsf{p}(S_{m^*})\right)\right) \tag{G.37}$$

$$\mathsf{r}(S_{m^*}) + \mathsf{r}(T_{m^*+1}) = \sum_{j=1}^{m^*+1} \mathsf{r}(T_j) \leq \bar{\mathsf{r}}\left(\widehat{\mathsf{G}}\left(\widehat{\mathcal{W}}, \sum_{j=1}^{m^*+1} \mathsf{p}(T_j)\right)\right) = \bar{\mathsf{r}}\left(\widehat{\mathsf{G}}\left(\widehat{\mathcal{W}}, \mathsf{p}(S_{m^*}) + \mathsf{p}(T_{m^*+1})\right)\right) \tag{G.38}$$

Take $\delta' = \delta - \mathsf{p}(S_{m^*})$. By Lemma G.0.14,

$$\mathsf{r}(S_{m^*}) + \mathsf{r}(\mathsf{G}(T_{m^*+1}, \delta')) \leq \bar{\mathsf{r}}(\widehat{\mathsf{G}}(\widehat{\mathcal{W}}, \mathsf{p}(S_{m^*}) + \delta')) = \bar{\mathsf{r}}(\widehat{\mathsf{G}}(\widehat{\mathcal{W}}, \delta)) \tag{G.39}$$

Finally, observe that

$$\mathsf{r}(\mathsf{G}(\mathcal{V}, \delta)) = \sum_{j=1}^{m^*} \mathsf{r}(T_j) + \mathsf{r}(\mathsf{G}(T_{m^*+1}, \delta')) = \mathsf{r}(S_{m^*}) + \mathsf{r}(\mathsf{G}(T_{m^*+1}, \delta')) \tag{G.40}$$

And we have the theorem. □

The proof of Theorem 5.2.6 is identical to that of Theorem 5.2.5 in their logical sequences. Again, for readers' convenience, Theorem 5.2.5 is copied below.

Theorem 5.2.6. *Let $\mathcal{V} = \{T_i, i \in [1..m]\}$ be a set of disjoint power intervals with $T_i < T_{i+1}$.*

Let $\widehat{\mathcal{W}}$ be an achieving set of $\mathsf{T}(\mathcal{V})$ *with*

$$\sum_{j=i}^{M} \mathsf{r}(T_j) \geq \bar{\mathsf{r}}\left(\widehat{\mathsf{L}}\left(\widehat{\mathcal{W}}, \sum_{j=i}^{M} \mathsf{p}(T_j)\right)\right), \quad \forall i \in [1..M] \tag{G.41}$$

Then for all $\delta \in [0, \mathsf{p}(\mathsf{T}(\mathcal{V}))]$,

$$\mathsf{r}(\mathsf{L}(\mathcal{V}, \delta)) \geq \bar{\mathsf{r}}(\widehat{\mathsf{L}}(\widehat{\mathcal{W}}, \delta)) \tag{G.42}$$

Lemma G.0.15. *Let $(a, a+p]$ be the combined user of $(a_1, a_1+p_1]$ and $(a_2, a_2+p_2]$. Let $(a_1', a_1'+p_1]$ and $(a_2', a_2'+p_2]$ have identical combined user $(a, a+p]$. Suppose $a_2' \geq a_2$, then for all $\delta \in [0, p_1]$*

$$\mathsf{r}((a_1+\delta, a_1+p_1]) + \mathsf{r}((a_2, a_2+p_2]) \geq \mathsf{r}((a_1'+\delta, a_1'+p_1]) + \mathsf{r}((a_2', a_2'+p_2]) \tag{G.43}$$

Proof. This lemma is equivalent to showing that

$$\mathsf{r}((a_1+\delta, a_1+p_1]) + \mathsf{r}((a_2, a_2+p_2]) \tag{G.44}$$

subject to

$$\mathsf{T}\left(\{(a_1, a_1+p_1], (a_2, a_2+p_2]\}\right) = (a, a+p] \tag{G.45}$$

is a decreasing function with the increase of a_2.

As in the proof of Lemma G.0.12, from (G.45),

$$\frac{a_1+p_1}{a_1}\frac{a_2+p_2}{a_2} = \frac{a+p}{a} \tag{G.46}$$

Let $c = \frac{a+p}{a}$,

$$a_1 = \frac{p_1}{\frac{ca_2}{a_2+p_2} - 1} \tag{G.47}$$

By the definition of $\mathsf{r}()$ function, there is the following for (G.44):

$$\mathsf{r}((a_1+\delta, a_1+p_1]) + \mathsf{r}((a_2, a_2+p_2]) = \frac{1}{2}\ln\left(\frac{a_1+p_1}{a_1+\delta}\frac{a_2+p_2}{a_2}\right) \tag{G.48}$$

Hence, we may equivalently show that the product

$$\frac{a_1 + p_1}{a_1 + \delta} \frac{a_2 + p_2}{a_2} \tag{G.49}$$

decreases with the increase of a_2 subject to condition (G.47).

Substitute (G.47) into (G.49),

$$\begin{aligned}
\frac{a_1 + p_1}{a_1 + \delta} \frac{a_2 + p_2}{a_2} &= \frac{\frac{p_1}{\frac{ca_2}{a_2+p_2}-1} + p_1}{\frac{p_1}{\frac{ca_2}{a_2+p_2}-1} + \delta} \frac{a_2 + p_2}{a_2} \\
&= \frac{p_1 \left(1 + \left(\frac{ca_2}{a_2+p_2} - 1\right)\right)}{p_1 + \delta \left(\frac{ca_2}{a_2+p_2} - 1\right)} \frac{a_2 + p_2}{a_2} \\
&= \frac{\frac{cp_1 a_2}{a_2+p_2}}{(p_1 - \delta) + \frac{ca_2\delta}{a_2+p_2}} \frac{a_2 + p_2}{a_2} \\
&= \frac{cp_1}{(p_1 - \delta) + \frac{ca_2\delta}{a_2+p_2}}
\end{aligned} \tag{G.50}$$

Observe that $\frac{a_1+p_1}{a_1+\delta} \frac{a_2+p_2}{a_2} \geq 0$ for all $\delta \in [0, p_1]$.

Taking the partial differential of $\frac{a_1+p_1}{a_1+\delta} \frac{a_2+p_2}{a_2}$ with respect to a_2.

$$\begin{aligned}
\frac{\partial}{\partial a_2} \left(\frac{a_1 + p_1}{a_1 + \delta} \frac{a_2 + p_2}{a_2} \right) &= -\frac{cp_1 \left(\frac{c\delta}{a_2+p_2} - \frac{ca_2\delta}{(a_2+p_2)^2}\right)}{\left((p_1 - \delta) + \frac{ca_2\delta}{a_2+p_2}\right)^2} \\
&= -\frac{cp_1 \frac{c\delta p_2}{(a_2+p_2)^2}}{\left((p_1 - \delta) + \frac{ca_2\delta}{a_2+p_2}\right)^2}
\end{aligned} \tag{G.51}$$

Since $\delta \leq p_1$, we have

$$\frac{\partial}{\partial a_1} \left(\frac{a_1 + p_1}{a_1 + \delta} \frac{a_2 + p_2}{a_2} \right) \leq 0 \tag{G.52}$$

This completes the proof. □

Lemma G.0.16. *For a given power interval T, let $\widehat{\mathsf{W}}$ be a simple set of time-power bi-intervals with power specification $\mathsf{p}(T)$. Suppose further that*

$$\mathsf{r}(T) \geq \bar{\mathsf{r}}(\widehat{\mathsf{W}}) \tag{G.53}$$

Then for all $\delta \in [0, \mathsf{p}(T)]$

$$\mathsf{r}(\mathsf{L}(T,\delta)) \geq \bar{\mathsf{r}}(\widehat{\mathsf{L}}(\widehat{\mathbb{W}},\delta)) \tag{G.54}$$

Proof. Without the loss of generality, take

$$\widehat{\mathbb{W}} = \{(a_i, a_i + \mathsf{p}(T)] \times (\alpha_{i-1}, \alpha_i], i \in [1..m]\} \tag{G.55}$$

with $0 = \alpha_0 \leq \alpha_1 \leq \ldots \leq \alpha_m = 1$.

Define

$$\widehat{\mathbb{W}}(x) = \{(a_i - x, a_i - x + \mathsf{p}(T)] \times (\alpha_{i-1}, \alpha_i], i \in [1..m]\} \tag{G.56}$$

Notice that

$$\bar{\mathsf{r}}(\widehat{\mathbb{W}}(0)) = \sum_{i=1}^{m} \frac{\alpha_i - \alpha_{i-1}}{2} \ln\left(\frac{a_i + \mathsf{p}(T)}{a_i}\right) = \bar{\mathsf{r}}(\widehat{\mathbb{W}}) \tag{G.57}$$

Therefore, by assumption,

$$\mathsf{r}(T) \geq \bar{\mathsf{r}}(\widehat{\mathbb{W}}(0)) \tag{G.58}$$

Observe that $\bar{\mathsf{r}}(\widehat{\mathbb{W}}(x))$ increases continuously to ∞ as x decreases to $\min\{a_i, i \in [1..M]\}$. Therefore, there exists an $x^* \geq 0$ such that

$$\mathsf{r}(T) = \bar{\mathsf{r}}(\widehat{\mathbb{W}}(x^*)) \tag{G.59}$$

In other words, T is achieved by $\widehat{\mathbb{W}}(x^*)$ simply.

By Corollary F.0.11, for all $\delta \in [0, \mathsf{p}(T)]$,

$$\mathsf{r}(\mathsf{L}(T,\delta)) \geq \bar{\mathsf{r}}(\widehat{\mathsf{L}}(\widehat{\mathbb{W}}(x^*),\delta)) \tag{G.60}$$

There is also

$$\bar{\mathsf{r}}(\widehat{\mathsf{L}}(\widehat{\mathcal{W}}(x^*), \delta)) = \sum_{i=1}^{m} \frac{\alpha_i - \alpha_{i-1}}{2} \ln \left(\frac{a_i - x^* + \mathsf{p}(T)}{a_i - x^* + \mathsf{p}(T) - \delta} \right) \geq \sum_{i=1}^{m} \frac{\alpha_i - \alpha_{i-1}}{2} \ln \left(\frac{a_i + \mathsf{p}(T)}{a_i + \mathsf{p}(T) - \delta} \right) = \bar{\mathsf{r}}(\widehat{\mathsf{L}}(\widehat{\mathcal{W}}, \delta)) \tag{G.61}$$

Combining (G.60) and (G.61), we have the desired inequality for this lemma. □

Lemma G.0.17. *Let T_1 and T_2 be two disjoint power intervals with $T_1 < T_2$. Let $\widehat{\mathcal{W}}$ be a set of simple time-power bi-intervals such that*

$$\mathsf{r}(T_1) \geq \bar{\mathsf{r}}(\widehat{\mathsf{L}}(\widehat{\mathcal{W}}, \mathsf{p}(T_1))) \tag{G.62}$$

$$\mathsf{r}(T_1) + \mathsf{r}(T_2) \geq \bar{\mathsf{r}}(\widehat{\mathsf{L}}(\widehat{\mathcal{W}}, \mathsf{p}(T_1) + \mathsf{p}(T_2))) \tag{G.63}$$

Then for any $\delta \in [0, \mathsf{p}(T_2)]$,

$$\mathsf{r}(T_1) + \mathsf{r}(\mathsf{L}(T_2, \delta)) \geq \bar{\mathsf{r}}(\widehat{\mathsf{L}}(\widehat{\mathcal{W}}, \mathsf{p}(T_1) + \delta)) \tag{G.64}$$

Proof. If $\mathsf{r}(T_2) \geq \bar{\mathsf{r}}(\widehat{\mathsf{L}}(\widehat{\mathcal{W}}, \mathsf{p}(T_1) + \mathsf{p}(T_2)))$, the lemma is trivially true.

Otherwise, let T_1^* and T_2^* be such that

$$\mathsf{p}(T_1^*) = \mathsf{p}(T_1) \tag{G.65}$$

$$\mathsf{p}(T_2^*) = \mathsf{p}(T_2) \tag{G.66}$$

$$\mathsf{r}(T_2^*) = \bar{\mathsf{r}}(\widehat{\mathsf{L}}(\widehat{\mathcal{W}}, \mathsf{p}(T_2))) \tag{G.67}$$

$$\mathsf{r}(T_1^*) + \mathsf{r}(T_2^*) = \mathsf{r}(T_1) + \mathsf{r}(T_2) \tag{G.68}$$

By Lemma G.0.15, we have for all $\delta \in [0, \mathsf{p}(T_2)]$

$$\mathsf{r}(T_2) + \mathsf{r}(\mathsf{L}(T_1, \delta)) \geq \mathsf{r}(T_2^*) + \mathsf{r}(\mathsf{L}(T_1^*, \delta)) \tag{G.69}$$

Thus the lemma is proven if we can show

$$\mathsf{r}(T_2^*) + \mathsf{r}(\mathsf{L}(T_1^*, \delta)) \geq \bar{\mathsf{r}}(\widehat{\mathsf{L}}(\widehat{\mathcal{W}}, \mathsf{p}(T_2) + \delta)) = \bar{\mathsf{r}}(\widehat{\mathsf{L}}(\widehat{\mathcal{W}}, \mathsf{p}(T_2^*) + \delta)) \tag{G.70}$$

Combining (G.63) and (G.68),

$$\mathsf{r}(T_1^*) + \mathsf{r}(T_2^*) \geq \bar{\mathsf{r}}(\widehat{\mathsf{L}}(\widehat{\mathcal{W}}, \mathsf{p}(T_1) + \mathsf{p}(T_2))) \tag{G.71}$$

Therefore

$$\mathsf{r}(T_1^*) \geq \mathsf{r}(\widehat{\mathsf{G}}(\widehat{\mathcal{W}}, \mathsf{p}(T_1^*) + \mathsf{p}(T_2^*))) - \mathsf{r}(T_2^*) \tag{G.72}$$

Substitute (G.67) into the above inequality to get

$$\begin{aligned} \mathsf{r}(T_1^*) &\geq \bar{\mathsf{r}}(\widehat{\mathsf{L}}(\widehat{\mathcal{W}}, \mathsf{p}(T_1^*) + \mathsf{p}(T_2^*))) - \bar{\mathsf{r}}(\widehat{\mathsf{L}}(\widehat{\mathcal{W}}, \mathsf{p}(T_2^*))) \\ &= \bar{\mathsf{r}}(\widehat{\mathsf{cmpl}}(\widehat{\mathsf{L}}(\widehat{\mathcal{W}}, \mathsf{p}(T_1^*) + \mathsf{p}(T_2^*)), \widehat{\mathsf{L}}(\widehat{\mathcal{W}}, \mathsf{p}(T_2^*)))) \\ &= \bar{\mathsf{r}}(\widehat{\mathsf{L}}(\widehat{\mathsf{cmpl}}(\widehat{\mathcal{W}}, \widehat{\mathsf{L}}(\widehat{\mathcal{W}}, \mathsf{p}(T_2^*))), \mathsf{p}(T_1^*))) \end{aligned} \tag{G.73}$$

By Lemma G.0.16, for all $\delta \in [0, \mathsf{p}(T_2)]$,

$$\begin{aligned} \mathsf{r}(\mathsf{L}(T_1^*), \delta) &\geq \bar{\mathsf{r}}(\widehat{\mathsf{L}}(\widehat{\mathsf{L}}(\widehat{\mathsf{cmpl}}(\widehat{\mathcal{W}}, \widehat{\mathsf{L}}(\widehat{\mathcal{W}}, \mathsf{p}(T_2^*))), \mathsf{p}(T_1^*)), \delta)) \\ &= \bar{\mathsf{r}}(\widehat{\mathsf{L}}(\widehat{\mathsf{cmpl}}(\widehat{\mathcal{W}}, \widehat{\mathsf{L}}(\widehat{\mathcal{W}}, \mathsf{p}(T_2^*))), \delta)) \end{aligned} \tag{G.74}$$

Add (G.67) and (G.74),

$$\begin{aligned} \mathsf{r}(T_2^*) + \mathsf{r}(\mathsf{L}(T_1^*), \delta) &\leq \bar{\mathsf{r}}(\widehat{\mathsf{L}}(\widehat{\mathcal{W}}, \mathsf{p}(T_2))) + \bar{\mathsf{r}}(\widehat{\mathsf{L}}(\widehat{\mathsf{cmpl}}(\widehat{\mathcal{W}}, \widehat{\mathsf{L}}(\widehat{\mathcal{W}}, \mathsf{p}(T_2^*))), \delta)) \\ &= \bar{\mathsf{r}}(\widehat{\mathsf{L}}(\widehat{\mathcal{W}}, \mathsf{p}(T_2^*))) + \bar{\mathsf{r}}(\widehat{\mathsf{L}}(\widehat{\mathsf{cmpl}}(\widehat{\mathcal{W}}, \widehat{\mathsf{L}}(\widehat{\mathcal{W}}, \mathsf{p}(T_2^*))), \delta)) \\ &= \bar{\mathsf{r}}(\widehat{\mathsf{L}}(\widehat{\mathcal{W}}, \mathsf{p}(T_2^*) + \delta)) \end{aligned} \tag{G.75}$$

This completes the proof. □

Proof of Theorem 5.2.5: Let $m^* \in [1..m]$ be such that

$$\sum_{j=m^*-1}^{m} \mathsf{p}(T_j) \leq \delta \leq \sum_{j=m^*}^{m} \mathsf{p}(T_j) \tag{G.76}$$

Take $S_{m^*} = \mathsf{T}(\mathsf{tail}(\mathcal{V}, m^* - 1))$. By assumption, there are

$$\mathsf{r}(S_{m^*}) = \sum_{j=m^*}^{m} \mathsf{r}(T_j) \geq \bar{\mathsf{r}}\left(\widehat{\mathsf{L}}\left(\widehat{\mathcal{W}}, \sum_{j=m^*}^{m} \mathsf{p}(T_j)\right)\right) = \bar{\mathsf{r}}\left(\widehat{\mathsf{L}}\left(\widehat{\mathcal{W}}, \mathsf{p}(S_{m^*})\right)\right) \tag{G.77}$$

$$\mathsf{r}(S_{m^*}) + \mathsf{r}(T_{m^*-1}) = \sum_{j=m^*-1}^{m^*} \mathsf{r}(T_j) \geq \bar{\mathsf{r}}\left(\widehat{\mathsf{L}}\left(\widehat{\mathcal{W}}, \sum_{j=m^*-1}^{m} \mathsf{p}(T_j)\right)\right) = \bar{\mathsf{r}}\left(\widehat{\mathsf{L}}\left(\widehat{\mathcal{W}}, \mathsf{p}(S_{m^*}) + \mathsf{p}(T_{m^*-1})\right)\right) \tag{G.78}$$

Take $\delta' = \delta - \mathsf{p}(S_{m^*})$. By Lemma G.0.17, there is

$$\mathsf{r}(S_{m^*}) + \mathsf{r}(\mathsf{L}(T_{m^*+1}, \delta')) \geq \bar{\mathsf{r}}(\widehat{\mathsf{L}}(\widehat{\mathcal{W}}, \mathsf{p}(S_{m^*}) + \delta')) = \bar{\mathsf{r}}(\widehat{\mathsf{L}}(\widehat{\mathcal{W}}, \delta)) \tag{G.79}$$

Finally, observe that

$$\mathsf{r}(\mathsf{L}(\mathcal{V}, \delta)) = \sum_{j=m^*}^{m} \mathsf{r}(T_j) + \mathsf{r}(\mathsf{L}(T_{m^*-1}, \delta')) = \mathsf{r}(S_{m^*}) + \mathsf{r}(\mathsf{L}(T_{m^*-1}, \delta')) \tag{G.80}$$

And we have the theorem. □

Bibliography

[1] R. Ahlswede. Multi-way Communication Channels. In *Proc. 2nd. Int. Symp. Info. Theory*, pages 103–135, Tsahkadsor, USSR, September 1971.

[2] Patrick P. Bergmans. A Simple Converse for Broadcase Channels with Additive White Gaussian Noise. *IEEE Trans. Info. Theory*, IT-20:279–280, March 1974.

[3] Patrick P. Bergmans and Thomas M. Cover. Cooperative Broadcasting. *IEEE Trans. Info. Theory*, IT-20:317–324, May 1974.

[4] A. B. Carleial. *On the Capacity of Multiple-terminal Communication Network*. PhD thesis, Stanford University, August 1975.

[5] Thomas M. Cover. An Achievable Rate Region for the Broadcast Channel. *IEEE Trans. Info. Theory*, IT-21:39–404, July 1975.

[6] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, New York, 1991.

[7] Robert G. Gallager. *Information Theory and Reliable Communication*. John Wiley & Sons, New York, New York, 1968.

[8] Robert G. Gallager. A Perspective on Multiaccess Channels. *IEEE Trans Info. Theory*, IT-31:124–142, March 1985.

[9] A. Grant, B. Rimoldi, R. Urbanke, and P. Whiting. Rate-Splitting Multiple Access for Discrete Memoryless Channels. *to be published on IEEE Trans Info. Theory*, 1999.

[10] Steven V. Hanly and David N. Tse. Multi-Access Fading Channels: Part 2: Delay-Limited Capacities. In *Proc. 1997. Int. Symp. Info. Theory*, Ulm, Germany, June 1997.

[11] H. Liao. *Multi Access Channels*. PhD thesis, Dept. of Elec. Eng., Univ. of Hawaii, 1972. also available as "A Coding Theorem for Multiple Access Communications", in *Proc. Int. Symp. Info. Theory.*, Asilomar, CA, 1972.

[12] Bixio Rimoldi and Rudiger Urbanke. A Rate-Splitting Approach to the Gaussian Multiple-Access Channels. *IEEE Trans. Info. Theory*, IT-42(2):364–375, March 1996.

[13] Claude E. Shannon. A Mathematical Theory of Communication. *Bell Syst. Tech. J.*, 27:379–423,623–656, July and October 1948.

[14] Claude E. Shannon. Two-Way Communication Channels. In *Proc. 4th Berkeley Symp. on Mathematical Statistics and Probability*, volume 1, pages 611–644, Berkeley, CA, 1961.

[15] Claude. E. Shannon, Robert G. Gallager, and E. R. Berlekamp. Lower Bounds to Error Probability for Coding in Discrete memoryless Channels. *Inform. Contr.*, 10:65–103,522–552, 1967.

[16] Li Shu. A New Perspective to AWGN Multi-Access Channels. In *Proc. 1997. Int. Symp. Info. Theory*, Ulm, Germany, June 1997.

[17] Edward C. van der Meulen. A Survey of Multi-Way Channels in Information Theory, 1961–1976. *IEEE Trans. Info. Theory*, IT-23, January 1977.

[18] Edmund M. Yeh and Robert G. Gallager. Achieving the Multiple Access Capacity Region via Projective Time Sharing. In *Proc. 1998. Int. Symp. Info. Theory*, Cambridge, mass, August 1998.