Distributed Wireless Utility Maximization via Fast Power Control

ZHANG, Jialiang

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Doctor of Philosophy
in
Information Engineering

The Chinese University of Hong Kong
July 2013

Abstract of thesis entitled:

Distributed Wireless Utility Maximization via Fast Power Control

Submitted by ZHANG Jialiang

for the degree of Doctor of Philosophy

at The Chinese University of Hong Kong in July 2013

This thesis develops a new theoretical and algorithmic framework for practical distributed power control in wireless networks. It proposes and investigates fast optimal distributed power control algorithms applicable to LTE as well as cognitive radio. The proposed algorithms beat the well-known Qualcomm’s load-spillage distributed power control algorithm in [HandeRanganChiangWu08] and the distributed weighted proportional S-

INR algorithm in [TanChiangSrikant11] in terms of both the optimality of the solution and the convergence speed.

Wireless network utility maximization via distributed power control is a classical and challenging issue that has attracted much research attention. The problem is often formulated as a system utility optimization problem under some transmit power constraints, where the system utility function is typically an increasing function of link signal-to-interference-plus-noise-ratio (SINR). This problem is complicated by the fact that these wireless devices may interfere with each other. In particular, the wireless devices are affected by each other's transmit power, and the transmit powers and interferences experienced by the devices are interwoven in a complex manner.

Despite that, there have been good centralized algorithms for solving the problem. "Decentralized" solutions, on the other hand, are a different story. In practice, decentralized algorithms in which the devices interact with each other in a loosely coupled manner to improve the network utility, are easier to deploy than centralized algorithms. However, the design of workable (and provably workable in the mathematical sense) solution is very challenging. Small neglects can lead to solutions that are invalid or non-convergent. For example, although both papers [HandeRanganChiangWu08] and [TanChiangSrikant11] claim their distributed algorithms to be optimal, we discover some experimental evidence suggesting that certain parts of these algorithms are not quite right. Oftentimes, the former fails to converge or converges extremely slowly, while the latter could diverge in the first few iterations.

To fix these glitches and to broaden the scope of the problem, we develop a new analytical and algorithmic framework with a more general formulation. With this framework, we can identify the sources of the defects and shortcomings of prior algorithms. We further construct an optimal distributed (sub)gradient projection algorithm with provably valid step size rules. Rig-

orous convergence proof and complexity analysis for our algorithm are given (note: convergence proof and complexity analysis were missing in [HandeRanganChiangWu08] and incorrect in [TanChiangSrikant11]). In some scenarios, our algorithm can be further accelerated to yield even better performance. Extensive simulation experiments confirm that our algorithms always outperform the prior algorithms, in terms of both optimality and efficiency. Specifically, simulation demonstrates at least 100 times faster convergence than the prior algorithms under certain scenarios.

In summary, this thesis solves the important SINR-based utility maximization problem and achieves significantly better results than existing work. It develops a new theoretical and algorithmic framework which completely addresses the difficult convergence and step-size issues. Going forward, we believe the foundation established in this work will open doors to other fast distributed wireless and mobile solutions to problems beyond

the power control problem addressed here.

论文题目：基于分布式快速功率控制的无线网络效用最大化

作者：张家良

学校：香港中文大学

学系：信息工程学系

修读学位：哲学博士

摘要

本论文开发出了一个全新的理论和算法框架用於无线网络的分布式功率控制。我们提出两种快速分布式功率控制算法，并对此作了深入的研究。此种算法相当普适，比如适用于目前热门的LTE和认知无线电网络。它在解的最优性以及收敛速度等方面击败了著名的高通公司的”荷载溢出型分布式功率控制算法”（收录于重要论文[HandeRanganChiangWu08]）以及”分布式加权比例型信干噪比均衡算法”（收录于重要论文[TanChiangSrikant11]）。

作为一个重要而富有挑战性的研究课题，通过分布式功率控制达至无线网络效用的最大化一直受到业界的普遍关

注。这方面的研究通常把问题表述为一个最优化问题，即在某些功率约束条件下，优化整体系统的效用函数。（其中，系统的效用函数通常是各无线收发链路的信干噪比的增函数。）此问题已经有了不错的集中式解决方案，但成本更低廉、更易于布置、更为实用的分布式解决方案则欠奉，尤其是经严格证明可行的分布式解决方案。这是因为分布式算法一般只适用于相对简单或者有特殊结构的优化问题。而无线设备之间的相互干扰和各自信号功率之间的复杂关系使得分布式求解极其困难。在算法设计上，很小的疏漏就可能导致解决方案无效或者不收敛。例如，尽管论文[HandeRanganChiangWu08]和[TanChiangSrikant11]都声称各自的分布式算法提供了问题的最优解，但我们通过大量的仿真实验以及理论研究发现并非如此。我们发现”荷载溢出型分布式功率控制算法”时常要么无法收敛，要么收敛得极其慢。而”分布式加权比例型信干噪比均衡算法”则经常在几次迭代之後就已经发散。

我们开发出了全新的分析和算法框架，并将其推广到适用

于一般线性功率约束的情况。（前述论文的分析框架是基于某些非常特殊的线性功率约束。）在此基础上，我们逐一找出了前述算法中的错漏之处，并设计出我们的分布式梯度投影功率控制算法，以及与之相匹配的步长规则。我们严格证明了该步长规则的有效性和算法的收敛性、最优性，并给出了算法复杂度的分析。（相较之下，[HandeRanganChiangWu08]在算法收敛性证明上语焉不详，在其它方面则付之阙如；而[TanChiangSrikant11]的算法收敛性证明存在明显错误，在其它方面同样付之阙如。）在某些情况下，我们的算法可以进一步提速并提升运行性能。大量的仿真实验证实我们的算法在解的最优性和运行速度两方面都较前述算法优越。在某些情况下，我们算法的收敛速度上百倍快于前述算法。

总而言之，本论文成功解决了重要的效用优化问题并取得比前述论文更好的结果。它开发出全新的理论和算法框架，完全解决了步长规则和收敛性、最优性这些难题。展望未来，我们相信，本论文为快速功率控制在无线和移动解决方案中的应用打下了坚实的理论基础。我们期待该理论框架能够提供更多

问题的解决方案。

Thesis/Assessment Committee

Professor YEUNG Wai-Ho (Chair)

Professor LIEW Soung-Chang (Thesis Supervisor)

Professor HUANG Jianwei (Committee Member)

Professor NAIR Chandra M. (Committee Member)

Professor LAU Vincent (External Examiner)

Acknowledgments

I would like to express my hearty gratitude to my supervisor, Professor LIEW Soung Chang for kindly providing valuable advice, direction and support throughout my postgraduate study. He guided me the first step into research, from revising a sentence in a paper, delivering a message in a presentation, to breaking an intractable problem naturally and intuitively into tractable pieces. His enthusiasm in research, his insight, and his efficiency is highly impressive. His words: "To be Newton and Einstein, to publish seminal work rather than incremental work" have been always encouraging me to keep on going forward in my research.

I would also greatly appreciate Professor Cheewei Tan, Professor Mung Chiang, Professor Minghua Chen, Professor Jianwei Huang, Professor Wing-Cheong Lau, Professor Sid Chau, Professor Angela Zhang, Professor Robert Li, Professor Anthony So, Professor Will Ng, Professor Raymond Yeung, Professor Dah-Ming Chiu, Professor Wing Wong, Professor Ken Ma and Professor Tony Lee for their helpful discussion and suggestion in my research. Sincere thanks are also given to all other professors in IE department, my colleagues, my friends, especially Ms. Dr. Liqun Fu, Dr. Caihong Kai, Dr. Lu Lu, Dr. Shengli Zhang, Mr. Jianghao He, Mr. Shen Feng, Mr.

Qing Yang, Mr. Taotao Wang, Mr. Lizhao You and Ms. Meng Wang, for their encouragements and help.

Finally, this thesis is dedicated to my family. Their endless love contributes to the source of my strength to face challenges and overcome difficulties.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Power control is an instrumental and fundamental technique to boost the performance and efficiency of wireless systems. It plays a critical role in many wireless communication systems, including LTE, femtocell, cognitive radio and heterogeneous network, just to name a few. It is a powerful mechanism for energy management, interference mitigation, connectivity management and system utility maximization. It adapts the transmit powers to combat the impairment due to channel fading and mutual interference to ensure reliable communication and good performance. Important system performance metrics like data rate, bit error rate and outage probability, are related to signal-to-interference-plus-noise-ratio (SINR), which in turn depends on power allocation.

In essence, power control is an exercise of resource sharing among communication links in the system. On one hand, a communication link must be allocated enough power to maintain a satisfactory SINR; on the other hand, the allocated power should not be too high to generate excessive interference to other links, or to violate power regulations like maximal transmitted power determined by the dynamic range of the power amplifier. Such resource allocation problem is often formulated as a system utility optimization problem under some power constraints, where the system utility function is typically an increasing function of link SINRs. This is the core problem that this thesis examines.

For scalability and deployment feasibility, we need a fast optimal power control algorithm that is amenable to distributed implementation, which turns out to be a very challenging problem. In general, there are two major sources of difficulties in distributed power control:

1. The lack of global coordination and information. Each link has to make its own decision and perform computation based on local measurements and local information exchange.
2. The difficulty arising from the problem structure. For example, a nonconvex problem is hard even for a centralized solver. And in practice, the system utility is usually nonconvex in power.

To get some insight on how the existing distributed power control algorithms handle these two difficulties, we first consider the Foschini-Miljanic algorithm [21], the most famous and widely implemented distributed power control algorithm. This algorithm is simple, elegant and insightful: each link just iteratively scales its power according to the ratio of its target SINR to its current SINR. The point is that the global information needed for each link is already summarized in the current SINR, which is locally measurable. However, as was pointed out in [25], the Foschini-Miljanic algorithm only works well in the non-power-constrained scenario with feasible and prefixed target SINR. An infeasible target SINR may keep driving the system to increase the overall power, resulting in instability and excessive mutual interference. The algorithm by itself does not provide any distributed feasibility check, nor a mechanism for the choice and update of the target SINRs. Therefore, it does not provide a satisfactory solution on its own.

A groundbreaking remedy, well-known as Qualcomm's load-spillage distributed power control algorithm, is proposed in the seminal work [25]. Assuming special box power constraints or box interference constraints, [25] first discovers an analytical framework to transform the nonconvex problem in power domain to a convex problem in SINR domain, which works for a broad spectrum of practical system utilities. This turns the power control problem into an "SINR control" problem, which turns out to have a distributedly computable ascent direction for a "suitable choice" of step size. Accordingly, the two-time-scale iterative load-spillage algorithm computes this ascent direction and updates the target SINRs in its large-time-scale iteration process; and for each and every target SINR update, runs a small-time-scale patched Foschini-Miljanic power control algorithm. The patch to the Foschini-Miljanic algorithm is suggested only for the special cases of box power constraint and box interference constraint: Each link is responsible for the feasibility check for its own constraint. If its constraint is violated, it will automatically “penalize” itself by reducing its own target SINR.

Although [25] claims its load-spillage algorithm to be optimal, we discover that oftentimes this claim may be invalid. Instead, the algorithm contains several drawbacks as follows:

1. It fails to provide a general valid step size rule, and a corresponding rigorous proof of convergence. Nor a complexity bound.
2. The “penalization” in the patched Foschini-Miljanic algorithm also subjects to a similar step-size-rule issue.
3. It only provides two empirical step-size rules. Simulation shows that, usually, applying these rules result in non convergence or very slow convergence. Therefore, they are by no mean “suitable choice” in general.

An alternative remedy is given by the algorithm 6 in the recent work [7]. It is quite similar to the load-spillage one, except that

1. The analytical framework of [7] assumes a special sum power constraint.
2. The large-time-scale SINR update in algorithm 6 is done by a gradient projection algorithm.
3. It fixes the small-time-scale Foschini-Miljanic algorithm by adding a normalization step, relying on the special structure of the sum power

constraint. This normalization ensures that the target SINRs will always return to the feasible region where the Foschini-Miljanic algorithm works well.

Coincidentally, we also disprove the claim of optimality in [7] and discover a number of its defects and errors:

1. Our analysis showed that its gradient projection algorithm does not generate ascent direction of the system utility in general, and its proof of convergence is incorrect. The complexity bound is also missing.
2. Simulation shows that, its algorithm 6 could diverge in the first few iterations.

Therefore, the optimal solution to the problems above was still pending after [25] and [7], and to our knowledge, no one had filled the gap until this thesis.

As will be shown, this thesis fills the gap, and solves a more general distributed SINR-based utility maximization problem with multiple linear power constraints applicable to LTE as well as cognitive radio. We construct a theoretical framework that simplifies the formulation and analysis of the core problem drastically. With the framework, we propose an optimal distributed (sub)gradient projection algorithm with provably valid step size rules. A thorough and rigorous convergence proof and complexity analysis are also given. We further identify the sources of the defects and several disadvantages of prior algorithms. Extensive simulation experiments demonstrated the superiority of our algorithm over prior algorithms in terms of both solution optimality and convergence speed. For instance, under certain scenarios, our algorithm achieves at least 100 times faster convergence. In the long term, we believe the algorithms and the theoretical framework established in this thesis will open doors to new distributed optimization techniques in other wireless communications problems beyond power control.

1.2 Thesis Organization

The remainder of this thesis is organized as follows:

Chapter 2 presents the system model, background knowledge of linear power constraints, network utility functions and problem formulation. Specifically, we show how to transform linear power constraints to SINR constraints, a key step to simplify our problem.

Chapter 3 shows some nice properties of the transformed SINR constraints that facilitate distributed fast optimization.

In **Chapter** 4, we establish optimality conditions, propose fast optimal distributed algorithms with proof of convergence and analysis of convergence rate. We also evaluate its performance and show its correctness and efficiency over prior algorithms via simulation experiments.

Chapter 5 compares our work with prior work and highlights our contributions.

Chapter 6 concludes this thesis and remarks on future research directions.

Appendix collects lengthy proofs and some subtle remarks.

1.3 Notations

The following notations are used. Boldface uppercase letters denote matrices, boldface lowercase letters denote column vectors, and italics denote scalars. For component-wise matrix (vector) comparison, $\mathbf{A} \geq \mathbf{B}$ ($\mathbf{u} \geq \mathbf{v}$) implies $\mathbf{A} - \mathbf{B}$ ($\mathbf{u} - \mathbf{v}$) is nonnegative, $\mathbf{A} > \mathbf{B}$ ($\mathbf{u} > \mathbf{v}$) implies each element of $\mathbf{A} - \mathbf{B}$ ($\mathbf{u} - \mathbf{v}$) is positive, and $\mathbf{A} \gneqq \mathbf{B}$ ($\mathbf{u} \gneqq \mathbf{v}$) implies at least one element of $\mathbf{A} - \mathbf{B}$ ($\mathbf{u} - \mathbf{v}$) is positive. For symmetric matrices $\mathbf{A}$ and $\mathbf{B}$, $\mathbf{A} \succeq \mathbf{B}$ implies $\mathbf{A} - \mathbf{B}$ is positive semi-definite. The Perron-Frobenius eigenvalue of a non-negative square matrix $\mathbf{B}$ is denoted by $\rho(\mathbf{B})$, and the Perron (right) and left eigenvectors of $\mathbf{B}$ associated with $\rho(\mathbf{B})$ are denoted by $\mathbf{x}(\mathbf{B})$ and $\mathbf{y}(\mathbf{B})$, respectively. The superscripts $(\cdot)^T$ and $(\cdot)^\dagger$ denote transpose and complex conjugate transpose, respectively. For an $L \times 1$ vector $\mathbf{x}$, we denote $e^{\mathbf{x}} := [e^{x_1}, \ldots, e^{x_L}]^T$, $\log \mathbf{x} := [\log x_1, \ldots, \log x_L]^T$, and $\mathbf{x}^{-1} := [\frac{1}{x_1}, \ldots, \frac{1}{x_L}]^T$. The diagonal matrix formed by the components of $\mathbf{x}$ is denoted by $diag(\mathbf{x})$. The element-wise product of $L \times 1$ vector $\mathbf{x}$ and $\mathbf{y}$ is denoted by $\mathbf{x} \circ \mathbf{y} := [x_1 y_1, \ldots, x_L y_L]^T$. For a positive integer L, denote by $\langle L \rangle$ the set $\{1, \ldots, L\}$. Let $P: X \to Y$ be a mapping from the space X to the space Y. For a subset $Z \subset X$, we denote by $P(Z)$ the image of Z. Finally, $\mathbf{e}_l$ denotes the lth unit coordinate vector, $\mathbf{I}$ denotes the identity matrix, and $\mathbf{0}$, $\mathbf{1}$ respectively denote the all zero and all one matrix (or vector) whose dimensions can be easily inferred from the context.

Chapter 2

System Model and Problem Formulation

2.1 System Model

Consider a general wireless system consisting of L logical unidirectional links labeled $1, \ldots, L$. We use nonnegative vector $\mathbf{p} = [p_1, \ldots, p_L]^T$ to denote the transmit power vector in which p_l is the transmit power of link l. We also use $L \times L$ positive matrix $\mathbf{G} = [G_{ij}]_{L\times L}$ and $L \times 1$ positive vector $\mathbf{n} = [n_1, \ldots, n_L]^T$ to specify the channel power gain and noise power, respectively. Specifically, G_{ij} is the channel power gain from the transmitter of the jth link to the receiver of the ith link, and n_l is the power of additive white Gaussian noise (AWGN) at the lth receiver. We adopt the conventional Gaussian interference channel model and assume a linear matched filter at each receiver (treating multiuser interference as AWGN). Then the Signal-to-Interference-plus-Noise Ratio (SINR) for the lth user is defined as the ratio of the received signal power $G_{ll}p_l$ to the cumulative interference plus noise power $\phi_l \triangleq \sum_{j \neq l, j \in \langle L \rangle} G_{lj}p_j + n_l$. We denote it by γ_l: $\gamma_l = G_{ll}p_l/\phi_l$. Let $\gamma := [\gamma_1, \dots, \gamma_L]^T$ and $\phi := [\phi_1, \dots, \phi_L]^T$. When G and n are given, γ is a functional mapping of p. To specify this mapping in a compact form, let $\mathrm{g} := [G_{11}, \dots, G_{LL}]^T$ be the vector formed by the main diagonal of G, and $\bar{\mathrm{G}} := \mathrm{G} - diag(\mathrm{g})$ be a punctured version of G by setting its main diagonal to zeros. Then $\gamma(\mathrm{p}) = \mathrm{g} \circ \mathrm{p} \circ \phi^{-1}$ where $\phi = \bar{\mathrm{G}}\mathrm{p} + \mathrm{n}$. On the other hand, to get the inverse mapping $\mathrm{p}(\gamma)$, we have

$$\mathrm{p} = diag(\gamma \circ \mathrm{g}^{-1})\bar{\mathrm{G}}\mathrm{p} + diag(\gamma \circ \mathrm{g}^{-1})\mathrm{n}, \tag{2.1}$$

from the defining equation of SINR. It is well-known that if $\gamma \in \Gamma := \{\gamma : \rho(diag(\gamma \circ \mathrm{g}^{-1})\bar{\mathrm{G}}) < 1, \gamma \geq 0\}$, (2.1) has a unique solution $\mathrm{p}(\gamma) = [\mathbf{I} - diag(\gamma \circ \mathrm{g}^{-1})\bar{\mathrm{G}}]^{-1} diag(\gamma \circ \mathrm{g}^{-1})\mathrm{n}$; otherwise, there is no solution. Hence, $\gamma(\mathrm{p}) : \mathbb{R}_+^L \rightarrow \Gamma$ and $\mathrm{p}(\gamma) : \Gamma \rightarrow \mathbb{R}_+^L$ are a pair of inverse mappings, which characterize the one-to-one correspondence between points in unconstrained power domain $\mathbb{R}_+^L$ and SINR domain Γ (see, e.g., Lemma 2.1 in [8]).

2.2 Nonnegative Linear Power Constraints

Power is limited by many practical concerns in wireless communication, and many of them can be modeled as a set of nonnegative linear power constraints. As an example, in cognitive radio, the secondary users are not allowed to generate excessive interference at certain frequency band to avoid affecting the operation of the primary network. To guarantee this, FCC proposed that interference-temperature constraints be imposed to regulate the secondary network [9]. Control stations of the primary network will monitor the cumulative interference from the secondary network and perform access control to make sure such interference is below a prescribed power threshold. Let p_l be the transmit power of the lth link in the secondary network, $G_{cs,l}$ be the channel power gain from the transmitter of lth link to a control station and P_{IT} be the threshold, then the interference-temperature constraint is a nonnegative linear power constraint $\sum_{l\in\langle L\rangle} G_{cs,l}p_l \leq P_{IT}$. Other popular examples include individual link power constraint $p_l \leq \bar{p}_l, \forall l \in \langle L\rangle$ [5, 25], sum power constraint $\mathbf{1}^T\mathbf{p} \leq \bar{P}$ [7] , individual link interference constraint $\sum_{j\neq l, j\in\langle L\rangle} G_{lj}p_j + n_l = \phi_l \leq \bar{\phi}_l, \forall l \in \langle L\rangle$ [25], just to name a few. We note that any nonnegative linear power constraint can be specified in the form of $\mathbf{c}^T\mathbf{p} \leq 1$ where $\mathbf{c} \gneq 0$. For instance, we have $\mathbf{c} = [G_{cs,1}/P_{IT}, \ldots, G_{cs,L}/P_{IT}]^T$ for the cognitive radio case, $\mathbf{c} = \mathbf{e}_l/\bar{p}_l$ for the individual link power constraint of link l, and $\mathbf{c} = \mathbf{1}/\bar{P}$ for the sum power constraint.

2.3 Network Utility

In this thesis, we consider the same family of network utility functions as in [7] and [25]. Every utility $U(\gamma)$ is strictly increasing with link SINRs γ. To enable distributed implementation, we assume that the utility is a product form or summation of separable terms of the SINRs of different receivers. Thus our network utility can be written as $U(\gamma) = \sum_{l\in\langle L\rangle} U_l(\gamma_l)$.

(For product form, just take logarithm.)

In this thesis, we limit our scope to convex optimization. For reasons that will become clear later, we assume that $U_l(\gamma_l)$'s are concave in $\log \gamma_l$. For analytical tractability, we also impose a mild condition that $U_l(\gamma_l)$'s are twice continuously differentiable. We notice that this utility family covers a broad spectrum of system performance metrics in practice. An example is the α-fair utility functions defined on the Shannon rate function $r(\gamma_l)$ with $\alpha \geq 1$ [25]. Shannon rate function $r(\gamma_l) = K_1 \log(1 + K_2\gamma_l)$ is of critical importance in digital communication. Different positive constants K_1 and K_2 can model Shannon capacity and realistic link rates under different bandwidth allocation schemes, modulations and BER requirements. When $\alpha = 1$, the utility represents proportionally fair link rate allocation: $U(\gamma) = \sum_{l\in\langle L\rangle} \log(r(\gamma_l))$. When $\alpha > 1$, the utility stands for α-fair link rate allocation: $U(\gamma) = \sum_{l\in\langle L\rangle}[r(\gamma_l)]^{1-\alpha}/(1-\alpha)$. Yet another example is the weighted proportionally fair SINR allocation $U(\gamma) = \sum_{l\in\langle L\rangle} w_l \log(\gamma_l)$. It is regarded as a good approximation to the weighted sum rate utility $U(\gamma) = \sum_{l\in\langle L\rangle} w_l \log(1 + \gamma_l)$ at the high SINR region.

2.4 Problem Formulation

The core of our thesis is to solve the following general network utility maximization problem with M nonnegative linear power constraints:

$$\begin{aligned} \max_{\mathrm{p}\geq 0} \quad & U(\gamma(\mathrm{p})) \\ \text{s.t.} \quad & \mathrm{c}_m^T\mathrm{p} \leq 1, \;\; \forall m \in \langle M \rangle \,. \end{aligned} \tag{2.2}$$

The primary hurdle is the nonconvexity of $U(\gamma(\mathrm{p}))$ in p, induced by the coupled mutual-interference terms in the denominator of the SINR. The prior work [7] and [25] investigated the special cases of problem (2.2) with specially chosen simple power constraints. The former considers a sum power constraint ($M = 1$, $\mathrm{c} = 1/\bar{P}$) while the latter a box power/interference constraint ($M = L$, $\mathrm{p} \leq \bar{\mathrm{p}}$ or $\phi \leq \bar{\phi}$). With the benefit from the special problem structure, they provide different angles of attack to make the problem convex. The former overcomes this hurdle by transforming the sum power constraint to a convex log-SINR constraint and optimizing in the log-SINR domain. The latter transforms the power/interference domain to the load/spillage domain. As will be shown later with our unifying framework, these two approaches are equivalent. Our transformation in principle follows [7], yet with a nontrivial generalization to resolve issues arising from the general linear power constraints. More importantly, we provide a correct fast optimal distributed solution to the transformed problem.

We commence our study by deriving the image of a single power constraint $\{\mathrm{p} : \mathrm{c}^T\mathrm{p} \leq 1, \mathrm{p} \geq 0\}$ in SINR domain, denoted by Γ_{c}. It is an entry point to further tackle the complicated case of multiple constraints $\{\mathrm{p} : \mathrm{c}_m^T\mathrm{p} \leq 1, \; \forall m \in \langle M \rangle, \mathrm{p} \geq 0\}$.

2.5 Characterization of $\Gamma_{\rm c}$

To characterize $\Gamma_{\rm c}$, it suffices to find the Pareto frontier of $\Gamma_{\rm c}$, denoted by $\bar{\Gamma}_{\rm c}$, thanks to the following monotonicity property of $\Gamma_{\rm c}$:

Lemma 1. *The function* $\mathrm{p}(\gamma)$ *is monotone on* $\Gamma_{\rm c}$*:*

$$\mathrm{p}(\gamma) \geq \mathrm{p}(\beta) \; if \; \gamma \in \Gamma_{\rm c} \; and \; \gamma \geq \beta \geq 0. \tag{2.3}$$

Equality holds if and only if $\gamma = \beta$*. Furthermore, the set* $\Gamma_{\rm c}$ *is monotonic with respect to the order* $\geq$*. That is, if* $\gamma \in \Gamma_{\rm c}$ *and* $\gamma \geq \beta \geq 0$ *then* $\beta \in \Gamma_{\rm c}$*.*

(For the proof and remark, please refer to Appendix A.)

Given $\bar{\Gamma}_{\rm c}$, the SINR region can then be represented by $\Gamma_{\rm c} = \{\gamma : \exists \beta \in \bar{\Gamma}_{\rm c}, \beta \geq \gamma\}$. More importantly, in many realistic scenarios where the network utilities are component-wisely increasing with γ , the optimal γ must locate on the Pareto frontiers. One can hence restrict the focus on $\bar{\Gamma}_{\rm c}$ without compromising optimality, as seen later. To figure out $\bar{\Gamma}_{\rm c}$, without loss of generality, we consider the case $\gamma > 0$. In the remainder of this section , we show an essential property that $\bar{\Gamma}_{\rm c}$ is the image of $\mathrm{c}^T\mathrm{p} = 1$, given by $\{\gamma : \; \rho(diag(\gamma \circ \mathrm{g}^{-1})(\bar{\mathrm{G}} + \mathrm{n}\mathrm{c}^T)) = 1\}$. Specifically, we will demonstrate that

1. $\mathrm{c}^T\mathrm{p}(\gamma) = 1$ is necessary for Pareto efficiency;
2. $\mathrm{c}^T\mathrm{p}(\gamma) = 1$ is equivalent to $\rho(diag(\gamma \circ \mathrm{g}^{-1})(\bar{\mathrm{G}} + \mathrm{n}\mathrm{c}^T)) = 1$; and
3. $\rho(diag(\gamma \circ \mathrm{g}^{-1})(\bar{\mathrm{G}} + \mathrm{n}\mathrm{c}^T)) = 1$ is also sufficient for Pareto efficiency.

Indeed, recall the definition of SINR that $\gamma_l = G_{ll}p_l/(\sum_{j\neq l, j\in\langle L\rangle} G_{lj}p_j + n_l), \forall l \in \langle L\rangle$. We have $\mathrm{p} > 0$ and $\mathrm{c}^T\mathrm{p} > 0$ from $\gamma > 0$. For any feasible p, if $\mathrm{c}^T\mathrm{p} < 1$, one can always proportionally increase the power p by a factor of $a = 1/\mathrm{c}^T\mathrm{p}$. Then $a\mathrm{p}$ is feasible ($\mathrm{c}^T a\mathrm{p} = 1$) and yields strongly Pareto dominant SINR performance $\gamma(a\mathrm{p}) > \gamma(\mathrm{p})$ ($\gamma_l(a\mathrm{p}) = G_{ll}p_l/(\sum_{j\neq l, j\in\langle L\rangle} G_{lj}p_j + n_l/a) > G_{ll}p_l/(\sum_{j\neq l, j\in\langle L\rangle} G_{lj}p_j + n_l) = \gamma_l(\mathrm{p}), \forall l \in \langle L\rangle$). Therefore $\mathrm{c}^T\mathrm{p} = 1$ is necessary for the Pareto efficiency.

To see the image of $\{\mathrm{p} : \mathrm{p} > 0, \mathrm{c}^T\mathrm{p} = 1\}$ in the SINR domain, we note the following lemma adapted from Theorem 6 and Lemma 8 in [1].

Lemma 2. *Let* $\mathbf{A}$ *be a nonnegative irreducible matrix, and* $\mathrm{b}, \mathrm{c} \gneq 0$ *two nonnegative vectors. Then the conditional eigenvalue problem*

$$\lambda\mathrm{p} = \mathbf{A}\mathrm{p} + \mathrm{b},\ \lambda \in \mathbb{R},\ \mathrm{p} \geq 0,\ \mathrm{c}^T\mathrm{p} = 1$$

has a unique solution $(\lambda_*, \mathrm{p}_*)$*, where* $\lambda_* = \rho(\mathbf{A} + \mathrm{bc}^T)$ *and* p_* *is the unique normalized Perron vector of* $\mathbf{A} + \mathrm{bc}^T$.

As a consequence, the equation $\mathrm{p} = (\mathbf{A}\mathrm{p} + \mathrm{b})/[\mathrm{c}^T(\mathbf{A}\mathrm{p} + \mathrm{b})]$ *has a unique nonnegative solution, which is* p_*.

We defer the proof and remark to Appendix B.

Apply this lemma to equation (2.1), by letting $\lambda = \lambda_* = 1$, $\mathbf{A}(\gamma) = diag(\gamma \circ \mathrm{g}^{-1})\bar{\mathbf{G}}$, and $\mathrm{b}(\gamma) = diag(\gamma \circ \mathrm{g}^{-1})\mathrm{n}$. Notice that $\gamma > 0$ and $\mathbf{G} > 0$ implies that $\mathbf{A}(\gamma) = diag(\gamma \circ \mathrm{g}^{-1})\bar{\mathbf{G}}$ is irreducible. So is $\mathbf{A}(\gamma) + \mathrm{b}(\gamma)\mathrm{c}^T = diag(\gamma \circ \mathrm{g}^{-1})(\bar{\mathbf{G}} + \mathrm{nc}^T)$. We can see that any positive γ in $\{\gamma : \rho(diag(\gamma \circ \mathrm{g}^{-1})(\bar{\mathbf{G}} + \mathrm{nc}^T)) = 1\}$

corresponds to a unique positive Perron vector of $diag(\gamma \circ g^{-1})(\bar{G}+nc^T)$, $p(\gamma)$, which satisfies $c^T p(\gamma) = 1$. On the other hand, for any positive p satisfying $c^T p = 1$, $\gamma(p) = g \circ p \circ (\bar{G}p + n)^{-1}$ must satisfy $\rho(diag(\gamma(p) \circ g^{-1})(\bar{G}+nc^T)) = 1$ and $\gamma > 0$. This establishes the one to one correspondence between $\{\gamma : \rho(diag(\gamma \circ g^{-1})(\bar{G} + nc^T)) = 1, \gamma > 0\}$ and $\{p : p > 0, c^T p = 1\}$. For brevity, we let B denote $diag(g^{-1})(\bar{G} + nc^T)$ in the following discussion.

Now we are ready to show that $c^T p = 1$ (or equivalently, $\rho(diag(\gamma)B) = 1$) is also sufficient for the Pareto efficiency in Γ_c. Suppose this is not the case, i.e., there exists a non Pareto efficient γ satisfying $\{\rho(diag(\gamma)B) = 1, \gamma > 0\}$. Then there also exists a β such that $\beta \gneqq \gamma > 0$ and $\beta \in \Gamma_c$. For any irreducible nonnegative matrix A, $\rho(A)$ is a strictly increasing function of the elements in A [10]. Hence $\rho(diag(\beta)B) > \rho(diag(\gamma)B) = 1$, which leads to $c^T p(\beta) \neq 1$. Since $\beta \in \Gamma_c$, we must have $c^T p(\beta) < 1$. However, $\beta \gneqq \gamma > 0$ implies $p(\beta) \gneqq p(\gamma) > 0$, which leads to a contradiction $1 = c^T p(\gamma) \leq c^T p(\beta) < 1$. This justifies the Pareto efficiency of $\rho(diag(\gamma)B) = 1$. In fact, for any $\gamma > 0$, let $a = \rho(diag(\gamma)B)$, then γ/a is on the Pareto frontier. Therefore, we have $\gamma \in \Gamma_c$ if $a \leq 1$ and $\gamma \notin \Gamma_c$ otherwise. To summarize, $\Gamma_c = \{\gamma : \rho(diag(\gamma)B) \leq 1\}$ for the case $\gamma > 0$.

The case where some components of γ are 0 is similar. We leave it in Appendix C and summarize the above discussion by the key theorem of this thesis.

Key Theorem: *The nonnegative linear power constraint* $c^T p \leq 1$ *is equivalent to the SINR constraint* $\rho(diag(\gamma)B) \leq 1$. *In particular,* $c^T p = 1$ *is equivalent to* $\rho(diag(\gamma)B) = 1$.

2.6 Multiple Constraints

Note the above discussion can be further generalized. Similar equivalence exists in the presence of multiple constraints, that is, feasible regions $\{\mathbf{p} : \max_{m\in\langle M\rangle}(\mathbf{c}_m^T\mathbf{p}) \leq 1\} = \bigcap_{m\in\langle M\rangle}\{\mathbf{p} : \mathbf{c}_m^T\mathbf{p} \leq 1\} \Longleftrightarrow \bigcap_{m\in\langle M\rangle}\Gamma_{\mathbf{c}_m} \triangleq \bigcap_{m\in\langle M\rangle}\{\gamma : \rho(diag(\gamma)\mathbf{B}_m) \leq 1\} = \{\gamma : \max_{m\in\langle M\rangle}\rho(diag(\gamma)\mathbf{B}_m) \leq 1\}$, and their corresponding Pareto frontiers $\{\mathbf{p} : \max_{m\in\langle M\rangle}(\mathbf{c}_m^T\mathbf{p}) = 1\} \Longleftrightarrow \{\gamma : \max_{m\in\langle M\rangle}\rho(diag(\gamma)\mathbf{B}_m) = 1\}$. By the same token, Lemma 2 can be readily enhanced:

Lemma 3. *Let* $\mathbf{A}$ *be a nonnegative irreducible matrix, and* $\mathbf{b}, \mathbf{c}_m \gneqq 0, \ \forall m \in \langle M\rangle$ *nonnegative vectors. Then the conditional eigenvalue problem*

$$\lambda\mathbf{p} = \mathbf{A}\mathbf{p} + \mathbf{b}, \ \lambda \in \mathbb{R}, \ \mathbf{p} \geq \mathbf{0}, \ \max_{m\in\langle M\rangle}(\mathbf{c}_m^T\mathbf{p}) = 1$$

has a unique solution $(\lambda_*, \mathbf{p}_*)$*, where* $\lambda_* = \max_{m\in\langle M\rangle}\rho(\mathbf{A} + \mathbf{b}\mathbf{c}_m^T)$ *and* $\mathbf{p}_*$ *is the unique normalized Perron vector of* $\mathbf{A} + \mathbf{b}\mathbf{c}_{m*}^T$*. Here* m_* *can be arbitrary element in the set* $\arg\max_{m\in\langle M\rangle}\rho(\mathbf{A} + \mathbf{b}\mathbf{c}_m^T)$*. Different choices of* m_* *(if applicable) result in the same normalized Perron vector* $\mathbf{p}_*$*.*

As a consequence, the equation $\mathbf{p} = (\mathbf{A}\mathbf{p}+\mathbf{b})/\max_{m\in\langle M\rangle}[\mathbf{c}_m^T(\mathbf{A}\mathbf{p} + \mathbf{b})]$ *has a unique nonnegative solution, which is* $\mathbf{p}_*$*.*

See Appendix D for the proof. It is worth pointing out that this enhanced lemma can help us in the sequel to cherry pick an active constraint from a set of constrains, which reduces the multi-constrained problems to easily solvable single-constrained problems.

Finally, we can reformulate our problem (2.2) in the SINR domain:

$$\begin{aligned} \max_{\gamma \geq 0} \quad & U(\gamma) \\ \textbf{s.t.} \quad & \rho(diag(\gamma)\mathbf{B}_m) \leq 1, \ \ \forall m \in \langle M \rangle \,, \end{aligned} \tag{2.4}$$

where $\mathbf{B}_m \triangleq diag(\mathbf{g}^{-1})(\bar{\mathbf{G}} + \mathbf{n}\mathbf{c}_m^T), \ \ \forall m \in \langle M \rangle$.

Chapter 3

Nice Properties of SINR Constraints

THE key theorem provides a way to transform network utility maximization problems from the power domain to the SINR domain. However, compared with the linear power constraint $\mathrm{c}^T\mathrm{p} \leq 1$, the SINR constraint $\rho(diag(\gamma)\mathrm{B}) \leq 1$ looks much more ugly and complicated. Worse still, its closed form is usually unavailable. One may doubt whether this transformation is helpful. We will remove this doubt by demonstrating some nice properties of this SINR constraint and show how they facilitate efficient distributed optimization. In the following, we will continue to focus on the all-active case where $\gamma > 0$. We suggest a change of variable $\gamma = e^{\hat{\gamma}}$ so that we can also operate in the log-SINR domain. Log-SINR vector is denoted by $\hat{\gamma} := [\hat{\gamma}_1, \ldots, \hat{\gamma}_L]^T$ where $\hat{\gamma}_l = \log \gamma_l, \forall l \in \langle L \rangle$. Then the log-SINR constraint becomes $\log \rho(diag(e^{\hat{\gamma}})\mathrm{B}) \leq 0$, and we use $\log \Gamma_c = \{\hat{\gamma} : \log \rho(diag(e^{\hat{\gamma}})\mathrm{B}) \leq 0\}$ and $\log \bar{\Gamma}_c = \{\hat{\gamma} : \log \rho(diag(e^{\hat{\gamma}})\mathrm{B}) = 0\}$ to denote the corresponding feasible log-SINR region and its Pareto frontier. We show that the function $f_{\mathrm{B}}(\hat{\gamma}) := \log \rho(diag(e^{\hat{\gamma}})\mathrm{B})$ is convex, differentiable and strictly increasing. Moreover, its gradient can be computed distributedly and efficiently, thanks to the intrinsic network-duality structure of $\mathrm{B} = diag(\mathrm{g}^{-1})(\bar{\mathrm{G}} + \mathrm{n}\mathrm{c}^T)$. This paves the way for the distributed optimization algorithm in next chapter.

3.1 Convexity, Differentiability and Monotonicity

Convexity is one of, if not, the most important properties in optimization. The convexity of $f_{\mathrm{B}}(\hat{\gamma})$ has been proved 30 years ago ([11]). Here we give an alternative "proof" by adding this new function to the cvx ([12]) library. So far, $f_{\mathrm{B}}(\hat{\gamma})$ has not been a standard library function in today's convex optimization tools. For others' convenience, we provide its source code in cvx, one of the most popular Matlab-based modeling systems for convex optimization. The idea is similar to that in section 4.5.4 of [19]. One can readily use this function by simply downloading and incorporating the file "logPFeig.m" ([11]) in the Matlab search path. We leave the details and explanation of this file to Appendix E. A direct consequence of convex $f_{\mathrm{B}}(\hat{\gamma})$ is that the feasible log-SINR region governed by any set of nonnegative linear power constraints must be a convex set. In addition, we can easily compute $\nabla f_{\mathrm{B}}(\hat{\gamma})$.

Lemma 4. $\nabla f_{\mathrm{B}}(\hat{\gamma}) = \mathrm{x}(diag(e^{\hat{\gamma}})\mathrm{B}) \circ \mathrm{y}(diag(e^{\hat{\gamma}})\mathrm{B})$, *where the right and left*

Perron eigenvectors are normalized such that $\mathrm{y}(diag(e^{\hat{\gamma}})\mathrm{B})^T\mathrm{x}(diag(e^{\hat{\gamma}})\mathrm{B}) = 1$.

Remark: The proof is given in Appendix F. An alternative proof is to apply the generalized Friedland-Karlin inequalities ([15]) to find out the unique supporting hyperplane to the convex set $f_{\mathrm{B}}(\hat{\gamma}) \leq 0$ at any point on its Pareto frontier, as shown in [8].

The strictly increasing monotonicity of $f_{\mathrm{B}}(\hat{\gamma})$ follows directly from the fact that the right and left Perron vector $\mathrm{x}(diag(e^{\hat{\gamma}})\mathrm{B})$ and $\mathrm{y}(diag(e^{\hat{\gamma}})\mathrm{B})$ are positive, which implies the positivity of $\nabla f_{\mathrm{B}}(\hat{\gamma})$. This again justifies that the Pareto frontier of $\log \Gamma_c$ is $\{\hat{\gamma} : \; f_{\mathrm{B}}(\hat{\gamma}) = 0\}$.

3.2 Fast Distributed Gradient Computation

Surprisingly interesting is the fact that the gradient $\nabla f_{\mathrm{B}}(\hat{\gamma}) = \mathrm{x}(diag(e^{\hat{\gamma}})\mathrm{B}) \circ \mathrm{y}(diag(e^{\hat{\gamma}})\mathrm{B})$ has a fast distributed computation algorithm inspired by network duality. This is realized in three steps:

1. Distributed power control to compute $\mathrm{x}(diag(e^{\hat{\gamma}})\mathrm{B})$ which is the "Pareto optimal primal power".
2. Distributed power control to compute $\mathrm{y}(diag(e^{\hat{\gamma}})\mathrm{B})$ which is the "Pareto optimal dual interference plus noise".
3. Normalization.

The following derivation is a non-trivial generalization from the special one in [7] assuming $M = 1$, $\mathrm{c} = 1/\bar{P}$ and $\mathrm{n} = 1$.

3.2.1 Distributed SINR-Driven Single-Constrained Power Control

First, to compute $\mathrm{x}(diag(e^{\tilde{\gamma}})\mathrm{B})$, recall that the application of Lemma 2 to equation (2.1) provides a fixed-point equation $\mathrm{p} = (\mathrm{Ap}+\mathrm{b})/[\mathrm{c}^T(\mathrm{Ap}+\mathrm{b})]$ that has a unique positive solution $\mathrm{p}_* = \mathrm{x}(diag(\beta)\mathrm{B})$. Here $\mathrm{A}(\beta) = diag(\beta \circ \mathrm{g}^{-1})\bar{\mathrm{G}}$, $\mathrm{b}(\beta) = diag(\beta \circ \mathrm{g}^{-1})\mathrm{n}$ and $\beta > 0$ is on Pareto frontier such that $\rho(diag(\beta)\mathrm{B}) = 1$. An algorithm can be designed as follows:

Algorithm 3.1 Centralized Single-Constrained Power Control

```
1. Input: $\beta$, G, n.
2. Initialization. Compute parameter $\mathrm{A} = diag(\beta\circ\mathrm{g}^{-1})\bar{\mathrm{G}}$ and $\mathrm{b} = diag(\beta\circ\mathrm{g}^{-1})\mathrm{n}$, and choose an initial $\mathrm{p}[0]$ such that $\mathrm{c}^T\mathrm{p}[0] > 0$.
3. Iteration. For $k = 0, 1, 2, \ldots$, compute $\mathrm{p}[k+1] \leftarrow (\mathrm{Ap}[k]+\mathrm{b})/[\mathrm{c}^T(\mathrm{Ap}[k]+\mathrm{b})]$ iteratively until $\{\mathrm{p}[k]\}$ converges to $\mathrm{p}_*$.
4. Output: $\mathrm{p}_*$.
```

The convergence of this algorithm is guaranteed by Lemma 5 below. For Pareto efficient $\beta > 0$, this algorithm will converge to $\mathrm{p}_* = \mathrm{p}(\beta)$. For other $\beta > 0$, let $\rho(diag(\beta)\mathrm{B}) = \lambda$. Then $\beta/\lambda > 0$ is Pareto efficient. Noting that

$$\frac{\mathrm{A}(\beta)\mathrm{p}+\mathrm{b}(\beta)}{\mathrm{c}^T[\mathrm{A}(\beta)\mathrm{p}+\mathrm{b}(\beta)]} = \frac{\mathrm{A}(\beta/\lambda)\mathrm{p}+\mathrm{b}(\beta/\lambda)}{\mathrm{c}^T[\mathrm{A}(\beta/\lambda)\mathrm{p}+\mathrm{b}(\beta/\lambda)]},$$

we have $\mathrm{p}_* = \mathrm{p}(\beta/\lambda)$. So we can interpret the algorithm input β as a "target SINR". The above algorithm then computes $\mathrm{p}_* = \mathrm{x}(diag(\beta)\mathrm{B})$ whose corresponding SINR $\gamma(\mathrm{p}_*)$ is proportional to β: $\gamma(\mathrm{p}_*) = \beta/\rho(diag(\beta)\mathrm{B})$. Note that $\gamma(\mathrm{p}_*)$ lies on the Pareto frontier. Indeed, it is the intersection of the Pareto frontier $\bar{\Gamma}_\mathrm{c}$ and the straight line passing through the origin and β. Intuitively, we can view $\gamma(\mathbf{p}_*)$ as the "projection" of β onto $\bar{\Gamma}_\mathrm{c}$. In other words, this algorithm computes the optimal solution $\mathbf{p}_*$ to the single-constrained weighted SINR balancing problem

$$\max_{\mathbf{p}:\mathbf{c}^T\mathbf{p}\leq 1,\mathbf{p}\gneqq\mathbf{0}} \min_{l} \frac{\gamma_l(\mathbf{p})}{\beta_l}.$$

The optimal value is $1/\rho(diag(\beta)\mathbf{B})$. Viewed in the log-SINR domain, the algorithm projects the target log-SINR vector $\hat{\beta} = \log\beta$ to its Pareto efficient correspondence $\hat{\beta} - \mathbf{1}\log\rho(diag(e^{\hat{\beta}})\mathbf{B})$. The projection movement is in $\mathbf{1}$ direction. Specifically in the case $L = 2$, it is a 45-degree line.

Importantly, the above algorithm has a distributed equivalence:

Algorithm 3.2 Distributed SINR-Driven Single-Constrained Power Control

1. Initialization. Each link $l \in \langle L\rangle$ sets its own target SINR β_l and chooses initial power $p_l[0]$.
2. Iteration. In $[k+1]$th iteration, each link l first updates its power $p_l[k+1] \leftarrow [\beta_l/\gamma_l(\mathbf{p}[k])]p_l[k]$, then normalizes $p_l[k+1] \leftarrow p_l[k+1]/\sum_{l\in\langle L\rangle} c_l p_l[k+1]$.
3. Termination. Iteration will not stop until convergence to $\mathbf{p}_*$ (or some prescribed stopping criteria are satisfied). Then every link l obtains its own $(\mathbf{p}_*)_l$ and $\mathbf{x}_l(diag(\beta)\mathbf{B}) = (\mathbf{p}_*)_l$.

Remark: In general, the weighted sum power $\sum_{l\in\langle L\rangle} c_l p_l[k+1]$ in the normalization step can be computed using the fast distributed gossip algorithm at each link [16]. In the special case like cognitive radio, the control station can simply measure the cumulative interference $(\sum_{l\in\langle L\rangle} c_l p_l[k+1])$ from the secondary network and broadcast.

This distributed algorithm is simple and insightful. In every iteration, each link first performs a Foschini-Miljanic distributed power control step [21] (i.e., to scales its power according to the ratio of target SINR to current SINR), and then re-scale its power again to meet the power constraint. Every normalization make sure that the resultant SINR is Pareto efficient. Better still, its convergence is guaranteed to be geometrically fast under a mild condition on the initial power, thanks to the following lemma adapted from Theorem 1 in [2].

Lemma 5. *Let $\tilde{T}$ be the mapping operator such that*

$$\tilde{T}\mathrm{p} = \frac{\mathrm{Ap} + \mathrm{b}}{\mathrm{c}^T(\mathrm{Ap} + \mathrm{b})},$$

where $\mathrm{A}(\beta) = diag(\beta \circ \mathrm{g}^{-1})\bar{\mathrm{G}}$, $\mathrm{b}(\beta) = diag(\beta \circ \mathrm{g}^{-1})\mathrm{n}$ *and* $\beta > 0$. *Then* $\lim_{k\to\infty} \tilde{T}^k \mathrm{p} = \mathrm{p}_*$ *for all* $\mathrm{p} : \mathrm{c}^T\mathrm{p} > 0$. *The convergence speed is geometrically fast.*

The proof and remark can be found in Appendix G.

To sum up, in the first step, we distributedly compute the Pareto efficient power vector $\mathrm{x}(diag(e^{\hat{\gamma}})\mathrm{B})$ by algorithm 1 with target SINR $\beta = e^{\hat{\gamma}}$. The resultant SINR vector is the "projection" of β onto $\bar{\Gamma}_c$, the gap is a factor of $1/\rho(diag(\beta)\mathrm{B})$.

3.2.2 Network Duality

The second step is to compute $\mathrm{y}(diag(\beta)\mathrm{B})$ from the target SINR β. Surprisingly, this is readily done by a slight modification of algorithm 2, owing to the nice network-duality properties. To see this realization, construct a dual network using the following rules adapted from [17]:

1. "Reverse the directions of all links." For each link, a primal transmitter (receiver) becomes a dual receiver (transmitter).
2. Replace the channel power gain matrix G by its transpose $\mathbf{G}^T$. If the channels are reciprocal ([18]), operation 1 already achieves this goal.

Practical examples of reciprocal wireless channels are those in time-division duplex (TDD) wireless systems, such as 802.11, 802.16, and TD-LTE. Throughout this thesis, for simplicity, we assume reciprocal channels unless otherwise specified. (The non-reciprocal case will be addressed in Remark 1 under Algorithm 3.3.) Then the dual network is simply constructed by switching the roles of transmitter and receiver in each link.

Between the primal and dual network we will see a beautiful symmetry. In the remainder of this section, for any parameter $(\mathbf{p}, \phi, \gamma, \mathbf{G}, \mathbf{g}, \bar{\mathbf{G}}, \mathbf{n}, \mathbf{c}$ or B) in primal network, we add a subscript d to denote its counterpart in the dual network. Then by definition, we have $\mathbf{G}_d = \mathbf{G}^T, \mathbf{g}_d = \mathbf{g}$ and $\bar{\mathbf{G}}_d = \bar{\mathbf{G}}^T$. Furthermore, we have the following useful network-duality lemma.

Lemma 6. *If* $\mathbf{c}_d\mathbf{n}_d^T = \mathbf{n}\mathbf{c}^T$*, then both primal and dual networks share the same SINR region (and Pareto frontier as well).*

Proof. Let $\mathbf{F} := \bar{\mathbf{G}} + \mathbf{n}\mathbf{c}^T$ and $\mathbf{F}_d := \bar{\mathbf{G}}_d + \mathbf{n}_d\mathbf{c}_d^T$. If $\mathbf{c}_d\mathbf{n}_d^T = \mathbf{n}\mathbf{c}^T$, then we have $\mathbf{F}_d = \mathbf{F}^T$. Note that $\rho(\mathbf{A}) = \rho(\mathbf{A}^T)$ and $\rho(\mathbf{D}\mathbf{A}^T) = \rho(\mathbf{A}\mathbf{D}) = \rho(\mathbf{D}\mathbf{A}\mathbf{D}\mathbf{D}^{-1}) = \rho(\mathbf{D}\mathbf{A})$ for any matrix A and invertible diagonal matrix D. Hence the SINR region of the dual network is $\{\gamma_d : 1 \geq \rho(diag(\gamma_d \circ \mathrm{g}_d^{-1})\mathbf{F}_d) = \rho(diag(\gamma_d \circ \mathrm{g}^{-1})\mathbf{F})\}$,which is the same as the primal network. □

Lemma 6 leads to many profound consequences. One of them is the fact that any Pareto optimal SINR vector $\gamma^\star$ has its one-to-one corresponding primal power $\mathrm{p}^\star := \mathrm{p}(\gamma^\star)$, primal cumulative interference plus noise $\phi^\star := \mathrm{g} \circ \mathrm{p}^\star \circ \gamma^{\star -1}$, dual power $\mathrm{p}_d^\star := \mathrm{p}_d(\gamma^\star)$ and dual cumulative interference plus noise $\phi_d^\star := \mathrm{g} \circ \mathrm{p}_d^\star \circ \gamma^{\star -1}$, given $\mathrm{c}_d \mathrm{n}_d^T = \mathrm{n}\mathrm{c}^T$. Here we add a superscript $\star$ to denote Pareto optimality of parameters. Surprisingly, they have rather simple and symmetric relationships. Define the diagonal matrix $\mathbf{D}^\star := diag(\gamma^\star \circ \mathrm{g}^{-1})$. From this definition, we have $\mathrm{p}^\star = \mathbf{D}^\star \phi^\star$ and $\mathrm{p}_d^\star = \mathbf{D}^\star \phi_d^\star$. At Pareto optimality, since $\mathrm{c}^T \mathrm{p}^\star = 1$ and $\mathrm{c}_d^T \mathrm{p}_d^\star = 1$, we also have $\phi^\star = \mathbf{F}\mathrm{p}^\star$ and $\phi_d^\star = \mathbf{F}_d \mathrm{p}_d^\star$. Then we can get $\phi_d^\star = \mathbf{F}_d \mathbf{D}^\star \phi_d^\star$ and hence $\phi_d^\star = \mathrm{x}(\mathbf{F}^T \mathbf{D}^\star) = \mathrm{y}(\mathbf{D}^\star \mathbf{F}) = \mathrm{y}(diag(\gamma^\star)\mathrm{B})$. Similarly, we can solve $\mathrm{p}^\star$, $\phi^\star$ and $\mathrm{p}_d^\star$ in terms of $\mathbf{F}$ and $\mathbf{D}^\star$. Their subtle relationships are summarized in Figure 3.1. Based on what we have intensively discussed in the former section, we can see the left Perron vector $\mathrm{y}(diag(\beta)\mathrm{B})$ is the Pareto efficient cumulative interference plus noise $\phi_d^\star$ corresponding to the target SINR β in the dual network, given $\mathrm{c}_d \mathrm{n}_d^T = \mathrm{n}\mathrm{c}^T$.

Now the question remains how to make sure $\mathrm{c}_d \mathrm{n}_d^T = \mathrm{n}\mathrm{c}^T$. A simple way to implement this is to force $\mathrm{c}_d = \mathrm{n}$ and $\mathrm{n}_d = \mathrm{c}$. However, in the real network, the noise vector at the dual receivers (i.e., the primal transmitters) n_d is not controllable and with high probability $\mathrm{n}_d \neq \mathrm{c}$. Luckily we have

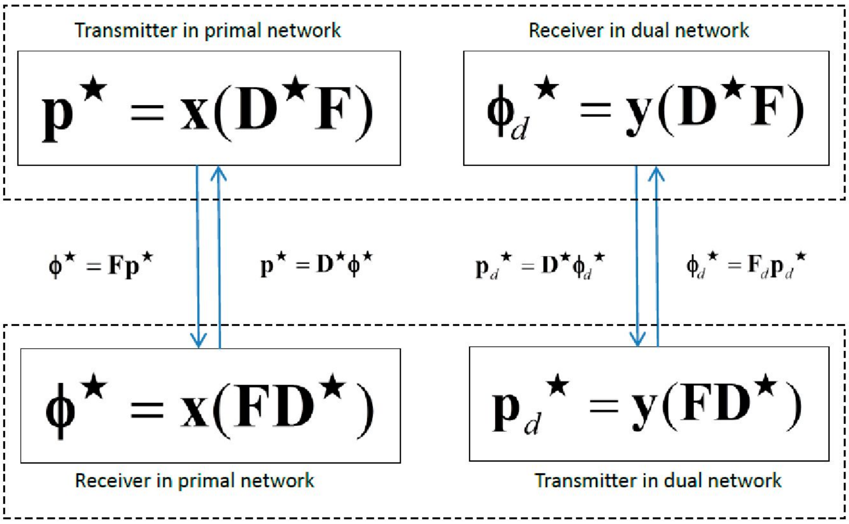
Transmitter in primal network
$\mathbf{p}^\star = \mathbf{x}(\mathbf{D}^\star\mathbf{F})$
Receiver in dual network
$\phi_d^\star = \mathbf{y}(\mathbf{D}^\star\mathbf{F})$
$\phi^\star = \mathbf{F}\mathbf{p}^\star$
$\mathbf{p}^\star = \mathbf{D}^\star\phi^\star$
$\mathbf{p}_d^\star = \mathbf{D}^\star\phi_d^\star$
$\phi_d^\star = \mathbf{F}_d\mathbf{p}_d^\star$
$\phi^\star = \mathbf{x}(\mathbf{F}\mathbf{D}^\star)$
Receiver in primal network
$\mathbf{p}_d^\star = \mathbf{y}(\mathbf{F}\mathbf{D}^\star)$
Transmitter in dual network

Figure 3.1: network duality

the following trick. For each dual link l, the cumulative interference plus noise it perceives is $\phi_{d,l} = \sum_{j\neq l, j\in\langle L\rangle} G_{d,lj}p_{d,j} + n_{d,l}$. Since link l can estimate $n_{d,l}$, it can extract the cumulative interference $\sum_{j\neq l,j\in\langle L\rangle} G_{d,lj}p_{d,j}$, compute a *virtual* cumulative interference plus noise $\phi_{v,l} = \sum_{j\neq l,j\in\langle L\rangle} G_{d,lj}p_{d,j} + c_l$ and scale its power accordingly. Then we have the following distributed algorithm:

Algorithm 3.3 Distributed Computation of $\mathrm{y}(diag(\beta)\mathrm{B})$ in Reciprocal Channels

1. Initialization. Reverse the link direction to construct the dual network. Each link $l \in \langle L\rangle$ sets its own target SINR β_l and chooses initial power $p_{d,l}[0]$.

2. Iteration. In $[k+1]$th iteration, each link l first computes its virtual cumulative interference plus noise from its perceived counterpart: $\phi_{v,l}(\mathrm{p}_d[k]) = \phi_{d,l}(\mathrm{p}_d[k]) + (c_l - n_{d,l})$, then updates its power $p_{d,l}[k+1] \leftarrow \beta_l\phi_{v,l}(\mathrm{p}_d[k])/g_l$, and finally normalizes $p_{d,l}[k+1] \leftarrow p_{d,l}[k+1]/\sum_{l\in\langle L\rangle} n_l p_{d,l}[k+1]$.

3. Termination. Iteration will not stop until convergence to p_{d*}. Then every link l obtains its own $(\phi_{v*})_l$, and $\mathrm{y}_l(diag(\beta)\mathrm{B}) = (\phi_{v*})_l$.

Remark:

1. For non-reciprocal channels, each link l can still use gossip algorithm to compute the cumulative interference $\sum_{j \neq l, j \in \langle L \rangle} G_{jl} p_{d,j}$, and then compute $\phi_{v,l}$. In this case, distributed implementation is still feasible, albeit complicated.

2. The above results of network duality and distributed power control are also applicable to MIMO beamforming networks. (See [7] for more details.) As a focus, this thesis will limit its scope to power-based SINR optimization without considering the optimization in beamformers.

Note that $\mathbf{x}(diag(\beta)\mathbf{B})$ and $\mathbf{y}(diag(\beta)\mathbf{B})$ computed in the first two steps are unique up to a positive scaling factor. The computation of $\nabla f_{\mathbf{B}}(\hat{\gamma}) = \mathbf{x}(diag(e^{\hat{\gamma}})\mathbf{B}) \circ \mathbf{y}(diag(e^{\hat{\gamma}})\mathbf{B}) / \mathbf{x}(diag(e^{\hat{\gamma}})\mathbf{B})^T \mathbf{y}(diag(e^{\hat{\gamma}})\mathbf{B})$ requires another normalization, which is easily done by invoking the gossip algorithm again.

As a summary to these two sections, we have shown that the function $f_{\mathbf{B}}(\hat{\gamma}) := \log \rho(diag(e^{\hat{\gamma}})\mathbf{B})$ is convex, strictly increasing and differentiable. Its gradient $\nabla f_{\mathbf{B}}(\hat{\gamma}) = \mathbf{x}(diag(e^{\hat{\gamma}})\mathbf{B}) \circ \mathbf{y}(diag(e^{\hat{\gamma}})\mathbf{B})$ can be computed distributedly and efficiently.

3.3 The Case of Multiple Constraints

We now consider the generalization to the multi-constrained feasible log-SINR region $\bigcap_{m \in \langle M \rangle} \log \Gamma_{\mathbf{c}_m} = \{\hat{\gamma} : \max_{m \in \langle M \rangle} \log \rho(diag(e^{\hat{\gamma}})\mathbf{B}_m) \leq 0\}$ and its Pareto frontier $\{\hat{\gamma} : \max_{m\in\langle M\rangle} \log \rho(diag(e^{\hat{\gamma}})\mathbf{B}_m) = 0\}$. Despite of the certain nondifferentiability introduced by the max operator, the case of multiple constraints largely inherits the nice properties of its single-constrained counterpart. For example, the function $\bar{f}(\hat{\gamma}) \triangleq \max_{m\in\langle M\rangle} \log \rho(diag(e^{\hat{\gamma}})\mathbf{B}_m)$ is still convex, and thus continuous, since $\max_{m\in\langle M\rangle}$ operator preserves the convexity of the component functions $\log \rho(diag(e^{\hat{\gamma}})\mathbf{B}_m)$, $\forall m \in \langle M\rangle$. Moreover, it is piecewise differentiable and always subdifferentiable. Let $AI(\hat{\gamma}) \triangleq \{m : f_{\mathbf{B}_m}(\hat{\gamma}) = \bar{f}(\hat{\gamma})\}$ denote the index set of "active" component functions at $\hat{\gamma}$. Then the subdifferential $\partial \bar{f}(\hat{\gamma})$ is the convex hull of $\{\nabla f_{\mathbf{B}_m}(\hat{\gamma}) | m \in AI(\hat{\gamma})\}$. In particular, if $AI(\hat{\gamma}) = \{m(\hat{\gamma})\}$ is a singleton set, then $\bar{f}(\hat{\gamma})$ is differentiable: $\nabla \bar{f}(\hat{\gamma}) = \nabla f_{\mathbf{B}_{m(\hat{\gamma})}}(\hat{\gamma})$. In practice, this happens with high probability. In case that $\bar{f}(\hat{\gamma})$ is nondifferentiable at $\hat{\gamma}$, then $AI(\hat{\gamma})$ must contain more than one element. Any gradient $\nabla f_{\mathbf{B}_m}(\hat{\gamma})$, $m \in AI(\hat{\gamma})$ of active component functions still serves as a subgradient of $\bar{f}(\hat{\gamma})$. Further, due to the positivity of $\{\nabla f_{\mathbf{B}_m}(\hat{\gamma})\ \forall m \in \langle M\rangle\}$, $\bar{f}(\hat{\gamma})$ is strictly increasing. These results are based on the well-known elementary convex optimization theory and can be found in, say, [14]. Of critical concern is whether efficient distributed (sub)gradient computation is still available. The answer is "Yes", thanks to the power-SINR equivalence relationship $\max_{m\in\langle M\rangle} (\mathbf{c}_m^T \mathbf{p}(\gamma)) \leq 1 \iff \max_{m\in\langle M\rangle} \rho(diag(\gamma)\mathbf{B}_m) \leq 1$, Lemma 3, and Lemma 7 given below.

Lemma 7. *Let $\tilde{T}$ be the mapping operator such that*

$$\tilde{T}\mathbf{p} = \frac{\mathbf{A}\mathbf{p} + \mathbf{b}}{\max_{m\in\langle M\rangle} \mathbf{c}_m^T(\mathbf{A}\mathbf{p} + \mathbf{b})},$$

where $\mathbf{A}(\beta) = diag(\beta \circ \mathbf{g}^{-1})\bar{\mathbf{G}}$, $\mathbf{b}(\beta) = diag(\beta \circ \mathbf{g}^{-1})\mathbf{n}$ *and* $\beta > 0$. *Then*

$\lim_{k\to\infty} \tilde{T}^k \mathrm{p} = \mathrm{p}_*$ *for all* $\mathrm{p} : \max_{m\in\langle M\rangle} (\mathrm{c}_m^T \mathrm{p}) > 0$. *The convergence speed is geometrically fast.*

Lemma 7 is the multi-constrained counterpart of Lemma 5. Their proofs are similar. Interested readers are referred to the remark in Appendix G.

Using Lemma 7 we can develop the multi-constrained counterpart of Algorithm 3.2.

Algorithm 3.4 Distributed SINR-Driven Multi-Constrained Power Control

1. Initialization. Each link $l \in \langle L\rangle$ sets its own target SINR β_l and chooses initial power $p_l[0]$.

2. Iteration. In $[k+1]$th iteration, each link l first updates its power $p_l[k+1] \leftarrow [\beta_l/\gamma_l(\mathrm{p}[k])]p_l[k]$, then normalizes $p_l[k+1] \leftarrow p_l[k+1]/\max_{m\in\langle M\rangle}(\sum_{l\in\langle L\rangle} c_{m,l}p_l[k+1])$.

3. Termination. Iteration will not stop until convergence to p_* (or some prescribed stopping criteria are satisfied). Then every link l obtains its own $(\mathrm{p}_*)_l$ and $\mathrm{x}_l(diag(\beta)\mathrm{B}_{m_*}) = (\mathrm{p}_*)_l$.

4. Interface to Algorithm 3.3. All links also agree on the final active constraint whose index is an arbitrary element m_* in the set $\arg\max_{m\in\langle M\rangle}(\mathrm{c}_m^T\mathrm{p}_*)$. Such constraint will be assigned to Algorithm 3.3: $\mathrm{c} \leftarrow \mathrm{c}_{m_*}$.

Remark: Again, in general, the weighted sum powers $\sum_{l\in\langle L\rangle} c_{m,l}p_l[k+1]\ \forall m \in \langle M\rangle$, their maximum in the normalization step and the final consensus on m_* can be computed using the fast distributed gossip algorithm. An extra $O(M)$ factor in complexity is expected over that of the single-constrained case.

Similarly, the above algorithm computes $\mathbf{p}_* = \mathbf{x}(diag(\beta)\mathbf{B}_{m_*})$ with target SINR β to get a projected Pareto efficient SINR

$$\gamma(\mathbf{p}_*) = \beta / \max_{m \in \langle M \rangle} \rho(diag(\beta)\mathbf{B}_m),$$

where $m_* \in AI(\log[\gamma(\mathbf{p}_*)])$. In other words, it computes the optimal solution $\mathbf{p}_*$ to the multi-constrained weighted SINR balancing problem

$$\max_{\mathbf{p}:\ \max_{m \in \langle M \rangle} (\mathbf{c}_m^T \mathbf{p}) \leq 1, \mathbf{p} \gneq \mathbf{0}} \min_l \frac{\gamma_l(\mathbf{p})}{\beta_l}.$$

The optimal value is $1/\max_{m \in \langle M \rangle} \rho(diag(\beta)\mathbf{B}_m)$. Viewed in the log-SINR domain, the algorithm projects the target log-SINR $\hat{\beta}$ to its Pareto efficient correspondence $\pi(\hat{\beta}) \triangleq \hat{\beta} - \mathbf{1} \max_{m \in \langle M \rangle} \log \rho(diag(e^{\hat{\beta}})\mathbf{B_m})$. The projection movement is still in $\mathbf{1}$ direction. (We will see in next chapter that $\mathbf{1}$ is a special direction which serves as the key of our network utility maximization algorithm.) Likewise, the corresponding distributed (sub)gradient computation is efficiently done by invoking both algorithms 3.4 and 3.3 (cf. Fig. 3.2). It returns the gradient of $\bar{f}(\hat{\gamma})$ once differentiable, and a subgradient otherwise.

As observed in all of our simulation experiments, the above algorithms 3.2, 3.3 and 3.4 converge very fast . For example, we observe that after 10 iterations, the maximal link-wise relative error of SINR

$$\max_{l \in \langle L \rangle} \left| \frac{\gamma_l(\mathbf{p}[10]) - \gamma_l(\mathbf{p}_*)}{\gamma_l(\mathbf{p}_*)} \right|$$

is bounded by a tiny constant, 0.12%. In other words, $\gamma(\mathbf{p}[10])$ is already

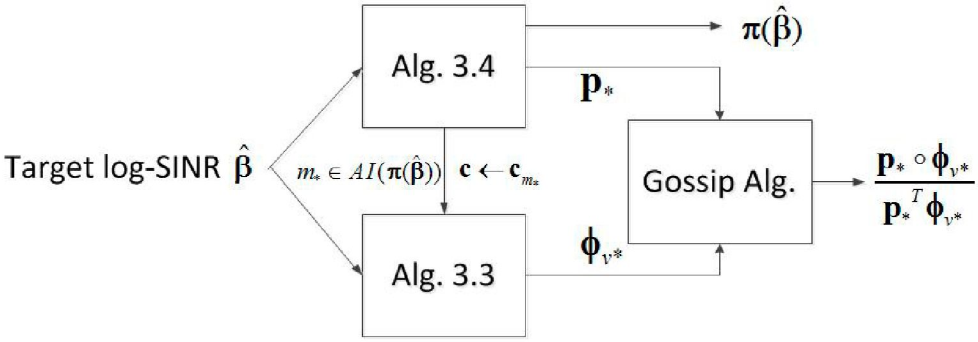
Alg. 3.4
$\pi(\hat{\boldsymbol{\beta}})$
$\mathbf{p}_*$
Target log-SINR $\hat{\boldsymbol{\beta}}$
$m_* \in AI(\pi(\hat{\boldsymbol{\beta}}))$
$\mathbf{c} \leftarrow \mathbf{c}_{m_*}$
Gossip Alg.
$\frac{\mathbf{p}_* \circ \boldsymbol{\phi}_{v*}}{\mathbf{p}_*^T \boldsymbol{\phi}_{v*}}$
$\boldsymbol{\phi}_{v*}$
Alg. 3.3

Figure 3.2: (Sub)gradient computation of $\bar{f}(\hat{\beta})$. A by product is the computation of the projection $\pi(\hat{\beta})$.

a fairly good approximation of $\gamma(\mathrm{p}_*)$ in practice. Hence, we will adopt this 10-iteration termination rule in our subsequent simulation.

Now we are ready to solve our core problem.

Chapter 4

Network Utility Maximization in Log-SINR Domain

IN what follows we will propose our algorithm to solve the network utility maximization problem (2.2), together with a simple convergence proof and complexity analysis. We will evaluate its performance by simulation experiments and demonstrate its advantages over algorithms in prior work [7] and [25].

We first show that our problem is a well-defined convex optimization problem with a unique solution. By transforming (2.4) to the log-SINR domain, our problem becomes

$$\begin{aligned} \max \quad & \hat{U}(\hat{\gamma}) \\ \text{s.t.} \quad & \log \rho(diag(e^{\hat{\gamma}})\mathbf{B}_m) \leq 0, \ \ \forall m \in \langle M \rangle \,, \end{aligned} \tag{4.1}$$

where $\mathrm{B}_m \triangleq diag(\mathrm{g}^{-1})(\bar{\mathrm{G}} + \mathrm{n}\mathrm{c}_m^T), \ \ \forall m \in \langle M \rangle$, and $\hat{U}(\hat{\gamma}) \triangleq U(e^{\hat{\gamma}})$. To specify our problem in a compact form, let $\hat{\rho}_m(\hat{\gamma}) \triangleq \log \rho(diag(e^{\hat{\gamma}})\mathrm{B}_m)$. Then (4.1) is rewritten as

$$\begin{aligned} \max \quad & \hat{U}(\hat{\gamma}) \\ \text{s.t.} \quad & \hat{\rho}_m(\hat{\gamma}) \leq 0, \ \ \forall m \in \langle M \rangle \,. \end{aligned}$$

Recall that our assumption on $U(\gamma)$ in section 2.3 is equivalent to that $\hat{U}(\hat{\gamma}) = \sum_{l \in \langle L \rangle} \hat{U}_l(\hat{\gamma}_l)$ is concave, twice continuously differentiable and strictly increasing with $\hat{\gamma}$, where $\hat{U}_l(\hat{\gamma}_l) \triangleq U_l(e^{\hat{\gamma}_l})$. Therefore our problem is a convex optimization problem. Based on the nice properties pertaining to the special direction 1, we show the following lemma.

Lemma 8. *The convex problem (4.1) has a unique global maximum and zero duality gap.*

Proof. For any realistic system, the power p is always bounded above, and the noise n is always positive and bounded away from 0. Therefore the SINR region is bounded. It is also closed (c.f. the "$\leq$" in the SINR constraints). From the well-known Weierstrass Theorem, a continuous $U(\gamma)$ can always attain its global maximum in the closed and bounded SINR region (in fact, on the Pareto frontier due to the strict monotonicity of $U(\gamma)$).

So can $\hat{U}(\hat{\gamma})$. This proves the existence of a global optimum. Consequently, any local maximum is also global, and the set of all global maxima is convex.

Next, we prove by contradiction that the global maximum is unique. Suppose there are two distinct global maxima $\hat{\beta}$ and $\hat{\beta}'$. Since they are Pareto optimal, we must have $\hat{\beta}' - \hat{\beta} \neq \alpha \mathbf{1}, \forall \alpha \in \mathbb{R}$. Then $\hat{\beta}'' = (\hat{\beta}' + \hat{\beta})/2$ must be a global maximum as well, due to the convexity of the global maxima set. However, note from [11] that $\hat{\rho}_m(\hat{\gamma})\ \forall m \in \langle M \rangle$ is strictly convex except in the direction $\mathbf{1}$. Then $\hat{\rho}_m(\hat{\beta}'') < [\hat{\rho}_m(\hat{\beta}') + \hat{\rho}_m(\hat{\beta})]/2 \leq 0\ \forall m \in \langle M \rangle$. $\hat{\beta}''$ is therefore strictly feasible rather than Pareto optimal, leading to a contradiction.

Finally, project $\hat{\beta} = \mathbf{0}$ to get its Pareto efficient correspondence $\pi(\mathbf{0}) = -\mathbf{1} \max\limits_{m \in \langle M \rangle} \log \rho(\mathbf{B_m})$. From Lemma 1, there are infinite many

$$\hat{\gamma} : \hat{\gamma} < -\mathbf{1} \max_{m \in \langle M \rangle} \log \rho(\mathbf{B_m})$$

that are strictly feasible, and hence the Slater's condition is valid. Strong duality and complementary slackness holds. The KKT conditions are both sufficient and necessary for primal-dual optimality. □

4.1 Single Active Constraint and Ascent Directions

Conventional line of analysis usually involves introducing an Lagrangian function $\mathcal{L}(\hat{\gamma}, \lambda) \triangleq \hat{U}(\hat{\gamma}) - \sum_{m \in \langle M \rangle} \lambda_m \hat{\rho}_m(\hat{\gamma})$ with multiple multipliers $\lambda :=$

$[\lambda_1, \ldots, \lambda_M]^T \geq 0$ to perform primal-dual iterations. As M grows, the excessive computational complexity of M-multiplier updates (which usually involves solving a bundle of nonlinear equations), however, is a curse to **distributed** implementation. The observation from our extensive simulation suggests that the number of active constraints is almost always one, when our iterate $\hat{\gamma}[t]$ glides along the Pareto frontier. This is the sparse structure to exploit.

To get some insight, we first consider the single-constraint case $M = 1$. We rewrite our problem to be

$$\begin{aligned} \max \quad & \hat{U}(\hat{\gamma}) \\ \text{s.t.} \quad & \hat{\rho}(\hat{\gamma}) \leq 0, \end{aligned}$$

where $\hat{\rho}(\hat{\gamma}) \triangleq \max\limits_{m \in \langle M \rangle} \hat{\rho}_m(\hat{\gamma})$. When $M = 1$, recall that $\hat{\rho}(\hat{\gamma})$ is differentiable everywhere: $\nabla \hat{\rho}(\hat{\gamma}) = \eta(\hat{\gamma})/\mathbf{1}^T \eta(\hat{\gamma})$, where $\eta(\hat{\gamma}) \triangleq \mathbf{x}(diag(e^{\hat{\gamma}})\mathbf{B}) \circ \mathbf{y}(diag(e^{\hat{\gamma}})\mathbf{B})$. We introduce a single-multiplier Lagrangian function $\mathcal{L}_\lambda(\hat{\gamma}) \triangleq \hat{U}(\hat{\gamma}) - \lambda \hat{\rho}(\hat{\gamma})$ and can write down the KKT equations as follows:

$$\begin{aligned} \nabla \mathcal{L}_\lambda(\hat{\gamma}) = \nabla \hat{U}(\hat{\gamma}) - \lambda \nabla \hat{\rho}(\hat{\gamma}) &= 0 \\ \hat{\rho}(\hat{\gamma}) &= 0 \\ \lambda &> 0. \end{aligned}$$

Note that $\mathbf{1}^T\nabla\hat{\rho}(\hat{\gamma}) = 1,\ \forall\hat{\gamma}$. Then at optimality,

$$\begin{aligned}
\lambda^\star &= \mathbf{1}^T\nabla\hat{U}(\hat{\gamma}^\star) \\
\nabla\hat{\rho}(\hat{\gamma}^\star) &= \frac{\nabla\hat{U}(\hat{\gamma}^\star)}{\mathbf{1}^T\nabla\hat{U}(\hat{\gamma}^\star)} \qquad (4.2)\\
\nabla\mathcal{L}_{\lambda^\star}(\hat{\gamma}^\star) &= 0.
\end{aligned}$$

Here we add a superscript $\star$ to denote optimal solutions. Now we are ready to introduce a simple gradient projection algorithm to solve this optimization problem.

Algorithm 4.1 (Sub)gradient Projected Network Utility Maximization

```
1. Initialization. Input an arbitrary Pareto efficient $\hat{\gamma}[0]$ and $\tilde{\nabla}\hat{\rho}(\hat{\gamma}[0])$.
   Compute $\lambda[0] \leftarrow \mathbf{1}^T\nabla\hat{U}(\hat{\gamma}[0])$.
2. Iteration. In $[t+1]$th iteration,
   (a) First compute $\tilde{\nabla}\mathcal{L}_{\lambda[t]}(\hat{\gamma}[t]) \leftarrow \nabla\hat{U}(\hat{\gamma}[t]) - \lambda[t]\tilde{\nabla}\hat{\rho}(\hat{\gamma}[t])$.
   (b) Then set a gradient step $\Delta\hat{\gamma}'[t] \leftarrow h[t]\tilde{\nabla}\mathcal{L}_{\lambda[t]}(\hat{\gamma}[t])$, and project $\hat{\gamma}[t]+$
       $\Delta\hat{\gamma}'[t]$ to the Pareto frontier to get $\hat{\gamma}[t+1]$ and $\tilde{\nabla}\hat{\rho}(\hat{\gamma}[t+1])$.
   (c) Compute $\lambda[t+1] \leftarrow \mathbf{1}^T\nabla\hat{U}(\hat{\gamma}[t+1])$. Repeat.
```

Remark:

1. As will be shown in the next section, this algorithm also works for the case of $M > 1$, where $\hat{\rho}(\hat{\gamma}) = \max\limits_{m\in\langle M\rangle}\hat{\rho}_m(\hat{\gamma})$ is piecewise differentiable and

always subdifferentiable. To be more accurate, here we use subgradient operator $\tilde{\nabla}$ in place of gradient ∇ where appropriate. For the current case of interest ($M = 1$), both operators are equivalent.

2. To get Pareto efficient $\hat{\gamma}[0]$ and $\tilde{\nabla}\hat{\rho}(\hat{\gamma}[0])$, one can invoke algorithms 3.4 and 3.3 .

3. $\lambda[t]$ can be computed using the fast distributed gossip algorithm.

4. $h[t]$ is the step size.

5. The gradient step and the projection step are orthogonal to each other: $\Delta\hat{\gamma}'[t] \perp -\mathbf{1}\hat{\rho}(\hat{\gamma}[t] + \Delta\hat{\gamma}'[t])$. To see this, note that $\mathbf{1}^T\nabla\mathcal{L}_{\lambda[t]}(\hat{\gamma}[t]) = 0$. Particularly when $L = 2$, the gradient moves along a 135-degree line, while the projection is 45-degree.

6. $\hat{\gamma}[t+1] = \pi(\hat{\gamma}[t] + \Delta\hat{\gamma}'[t]) = \hat{\gamma}[t] + \Delta\hat{\gamma}'[t] - \mathbf{1}\hat{\rho}(\hat{\gamma}[t] + \Delta\hat{\gamma}'[t])$.

7. Thanks to the projection step, our iterate $\hat{\gamma}[t]$ is always Pareto efficient. Then we always have $\mathcal{L}_\lambda(\hat{\gamma}[t]) = \hat{U}(\hat{\gamma}[t])$, regardless of λ's value. On the Pareto frontier, maximizing $\hat{U}(\hat{\gamma})$ is equivalent to maximizing $\mathcal{L}_\lambda(\hat{\gamma})$. This explains the reason why the gradient is proportional to $\nabla\mathcal{L}_{\lambda[t]}(\hat{\gamma}[t])$. Indeed, the projected gradient $\Delta\hat{\gamma}[t] \triangleq \hat{\gamma}[t+1] - \hat{\gamma}[t] = \Delta\hat{\gamma}'[t] - \mathbf{1}\hat{\rho}(\hat{\gamma}[t] + \Delta\hat{\gamma}'[t])$ is an ascent direction with a proper step size $h[t]$, which leads to the following nice convergence property.

Lemma 9. *Algorithm 4.1 converges with appropriately chosen step size $\{h[t]\}$, under the mild assumption that the first and second order derivatives of $\hat{U}$* and $\hat{\rho}$ are bounded:

$$\mathbf{1}^T\nabla\hat{U}(\hat{\gamma}) \leq V_\lambda,\ \nabla^2\hat{U}(\hat{\gamma}) \succeq -V_U\mathbf{I},\ \nabla^2\hat{\rho}(\hat{\gamma}) \preceq V_\rho\mathbf{I}, \forall\hat{\gamma}$$

where V_λ, V_U and V_ρ are some positive constants.

Here is the sketch of the proof. The key step is to show that under mild conditions, the projected gradient $\Delta\hat{\gamma}'[t] - 1\hat{\rho}(\hat{\gamma}[t] + \Delta\hat{\gamma}'[t])$ is an ascent direction with a proper step size $h[t]$. Specifically, we show that

$$\hat{U}(\hat{\gamma}[t+1]) - \hat{U}(\hat{\gamma}[t]) \geq ||\nabla\mathcal{L}_{\lambda[t]}(\hat{\gamma}[t])||^2\{h[t] - \kappa h[t]^2/2\}, \qquad (4.3)$$

where $\kappa = V_\lambda V_\rho + V_U(L+1)$. Then let $h[t] = 1/\kappa$, we have $\hat{U}(\hat{\gamma}[t+1]) - \hat{U}(\hat{\gamma}[t]) \geq ||\nabla\mathcal{L}_{\lambda[t]}(\hat{\gamma}[t])||^2/(2\kappa)$. Note that $\hat{U}(\hat{\gamma}^\star) - \hat{U}(\hat{\gamma}[0]) \geq \hat{U}(\hat{\gamma}[T+1]) - \hat{U}(\hat{\gamma}[0]) \geq \sum_{t=0}^{T} ||\nabla\mathcal{L}_{\lambda[t]}(\hat{\gamma}[t])||^2/(2\kappa)$, as a consequence, $||\nabla\mathcal{L}_{\lambda[t]}(\hat{\gamma}[t])|| \to 0$ as $t \to \infty$. Thanks to the strong duality proved in Lemma 8, this implies $\hat{\gamma}[t] \to \hat{\gamma}^\star$as $t \to \infty$. The proof of (4.3) is given in the Appendix H.

We notice that the network utility series $\{\hat{U}(\hat{\gamma}[t])\}$ is strictly monotonically increasing over time and converges eventually. Inspired by propositions 6.9.1 and 6.9.2 in [24], we may make a guess that our algorithm 4.1 has an iteration bound $t = O(\delta^{-1})$ for convergence to a δ-suboptimal solution $\hat{\gamma}[t] : \hat{U}(\hat{\gamma}[t]) \geq \hat{U}(\hat{\gamma}^\star) - \delta$. Adding a deliberately chosen extrapolation step to our projected gradient step is expected to further reduce the number of iterations to $t = O(\delta^{-1/2})$. This guess turns out to be true, and reasons will become clear in section 4.3.

We also want to emphasize that without a **provably valid** step size rule, convergence can not be guaranteed even if one always operates with ascent gradient directions. This is actually a missing part in the prior work

[7] and [25]. An appropriate step size rule is especially crucial for distributed implementation, since we intentionally avoid those adaptive yet complicated step size rules such as Armijo rule, Goldstein rule and (limited) minimization rule [23], which require centralized computation or excessive effort of global information exchange. (Note that each link may not even know whether the overall network utility is improving or not.) In practice, simple step size rules like diminishing step size rule and constant step size rule are mostly used. The key for a valid constant step size rule $h[t] = h$ is to ensure the monotonic improvement of $\{\hat{U}(\hat{\gamma}[t])\}$. One needs to carefully choose the step size $h \in (0, 1/\{V_\lambda V_\rho + V_U(L+1)\}]$. However, V_λ, V_U and V_ρ may not be known beforehand in reality. One may want to do some field experiments to estimate these parameters in advance, and conservatively select small enough $\{h[t]\}$. Smaller step sizes will probably result in slower convergence. In contrast, a properly chosen diminishing step size rule (say, $h[t] = h/\sqrt{t+1}$) can eventually zoom into the targeted interval $(0, 1/\{V_\lambda V_\rho + V_U(L+1)\}]$ after some initial attempts. As will be shown shortly, under a milder condition, the diminishing step size rule $h[t] = h/\sqrt{t+1}$ ensures universal convergence to the global optimum, in the sense that $\limsup_{t\to\infty} \hat{U}(\hat{\gamma}[t]) = \hat{U}(\hat{\gamma}^\star)$.

4.2 Multiple Constraints and Subgradient Projection

A careful examination on the derivation of algorithm 4.1 shows that it is by and large applicable to the multiple-constraint case. Recall that

$\hat{\rho}(\hat{\gamma}) = \max_{m \in \langle M \rangle} \hat{\rho}_m(\hat{\gamma})$ is differentiable almost everywhere except at those singular points where the active index set $AI(\hat{\gamma})$ is not a singleton. Therefore, if at optimality only one constraint is active, the optimality conditions (4.2) are still valid. The optimality conditions (4.2) may only become invalid where $AI(\hat{\gamma}^\star)$ is not a singleton. In this case, $\nabla\hat{\rho}(\hat{\gamma}^\star)$ does not exist and a single Lagrangian multiplier fails to capture the optimality. This is the first problem introduced by the multiple constraints.

Another problem, from the scenario where $\hat{\gamma}[t]$ happens to be a singular point, is that $\tilde{\nabla}\mathcal{L}_{\lambda[t]}(\hat{\gamma}[t]) = \nabla\hat{U}(\hat{\gamma}[t]) - \lambda[t]\tilde{\nabla}\hat{\rho}(\hat{\gamma}[t])$ is a subgradient of $\mathcal{L}_{\lambda[t]}(\hat{\gamma}[t])$, rather than a gradient. (Here we use $\tilde{\nabla}$ in place of ∇ to emphasize the difference between subgradient and gradient.) It is well-known that a subgradient does not need to be an ascent direction.

The worst is that we may no longer have a constant step size that is uniformly bounded: $h[t] = 1/[V_\lambda V_\rho + V_U(L+1)]$, even if we replace the mild condition $\nabla^2\hat{\rho}(\hat{\gamma}) \preceq V_\rho\mathbf{I}, \forall\hat{\gamma}$ in Lemma 9 with $\nabla^2\hat{\rho}_m(\hat{\gamma}) \preceq V_\rho\mathbf{I}, \forall m \in \langle M \rangle, \forall\hat{\gamma}$. What happens is that upon the projected gradient move, our iterate may transit from one active constraint to another. In other words, we may have $m_*(\hat{\gamma}[t]) \notin AI(\hat{\gamma}[t+1])$, where $m_*(\hat{\gamma}[t])$ is the choice of active constraint index when computing $\tilde{\nabla}\hat{\rho}(\hat{\gamma}[t])$. And $m_*(\hat{\gamma}[t]) \in AI(\hat{\gamma}[t+1])$ is critical for (4.3) to be valid. For a nonsingular $\hat{\gamma}[t]$, although one can always choose the step size to be small enough such that $\hat{\gamma}[t]$ and $\hat{\gamma}[t+1]$ are with the same active constraint, but we have no idea how small is enough. A sufficient improvement at each iteration such that $\hat{U}(\hat{\gamma}[t+1]) - \hat{U}(\hat{\gamma}[t]) \geq ||\nabla\mathcal{L}_{\lambda[t]}(\hat{\gamma}[t])||^2/(2\kappa)$ may no longer be valid.

The first two problems are not essential, and rarely occur in practice, because $\hat{\rho}(\hat{\gamma})$ is differentiable almost everywhere, and $\hat{U}(\hat{\gamma})$ is twice continuously differentiable. An arbitrarily small random perturbation on $\hat{\gamma}$ can recover differentiability with high probability. And even if $||\nabla\mathcal{L}_\lambda(\hat{\gamma}^\star)||$ does not exist, for the nonsingular $\hat{\gamma} \to \hat{\gamma}^\star$, we still have $||\nabla\mathcal{L}_\lambda(\hat{\gamma})|| \to 0$. The last problem is critical and challenging, because constant step size is no longer provably valid, even if V_λ, V_U and V_ρ is known.

We now provide a remedy to these problems in one shot. The key is to understand the true meaning of the "subgradient" and "projection" in our algorithm 4.1. Distinct from the classic subgradient projection algorithms (see [22, 23, 24]), our "projection" $\pi(\hat{\gamma}) = \hat{\gamma} - 1\hat{\rho}(\hat{\gamma})$ is along the special direction 1 to the Pareto frontier, rather than an Euclidean projection on the closed convex feasible region

$$EP(\hat{\gamma}) \triangleq \arg \min_{\hat{\rho}(\hat{\beta}) \leq 0} \left\|\hat{\gamma} - \hat{\beta}\right\|_2 .$$

The traditional subgradient projection method relies on the non expansion property of the Euclidean projection

$$\left\|EP(\hat{\gamma}) - EP(\hat{\beta})\right\|_2 \leq \left\|\hat{\gamma} - \hat{\beta}\right\|_2$$

to show that the distance of the current iterate to $\hat{\gamma}^\star$ is reduced with a proper step size. The non expansion property, however, does not come along with our "projection". In fact, we have

$$\left\|\pi(\hat{\gamma}) - \pi(\hat{\beta})\right\|_2 = \left\|\hat{\gamma} - \hat{\beta} - 1[\hat{\rho}(\hat{\gamma}) - \hat{\rho}(\hat{\beta})]\right\|_2 > \left\|\hat{\gamma} - \hat{\beta}\right\|_2 ,$$

if $\hat{\gamma} - \hat{\beta} \perp \mathbf{1}$ and $\hat{\gamma} \neq \hat{\beta}$. Surprisingly interesting is that there is a way to show almost the same thing.

Consider the special direction $\mathbf{1}$. It spans a subspace of $\mathbb{R}^L$ $\{\hat{\gamma} : \hat{\gamma} = \alpha\mathbf{1}, \alpha \in \mathbb{R}\}$, with an orthogonal complement $\mathbf{1}^{\perp} \triangleq \{\hat{\gamma} : \mathbf{1}^T\hat{\gamma} = 0\}$. We discover that there is a bijective mapping between the Pareto frontier and $\mathbf{1}^{\perp}$, the hyperplane passing through the origin, as depicted below. In fact, our algorithm 3.4 maps any point $\hat{\gamma}$ in $\mathbb{R}^L$ (including those on $\mathbf{1}^{\perp}$) to its unique Pareto efficient counterpart $\pi(\hat{\gamma}) = \hat{\gamma} - \mathbf{1}\max_{m \in \langle M \rangle} \log \rho(diag(e^{\hat{\gamma}})\mathbf{B_m})$. Conversely, any point on the Pareto frontier finds its way to $\mathbf{1}^{\perp}$ by simply conducting an orthogonal projection $\sigma(\hat{\gamma}) :\ \mathbb{R}^L \to \mathbb{R}^L$, where $\sigma(\hat{\gamma}) \triangleq [\mathbf{I} - \mathbf{1}\mathbf{1}^T/L]\hat{\gamma}$.

Note that in the subgradient case, we still have $\mathbf{1}^T\tilde{\nabla}\mathcal{L}_{\lambda[t]}(\hat{\gamma}[t]) = 0$. Then each iteration corresponds to a subgradient moving "horizontally" along $\mathbf{1}^{\perp}$, followed by a projection moving "vertically" along direction $\mathbf{1}$ to return to the Pareto frontier. Therefore, $\hat{\gamma}[t] \to \hat{\gamma}^{\star}$ if and only if $\sigma(\hat{\gamma}[t]) \to \sigma(\hat{\gamma}^{\star})$. In other words, the iterate $\hat{\gamma}[t]$ slides along the Pareto frontier approaching $\hat{\gamma}^{\star}$ is equivalent to its "shadow" $\sigma(\hat{\gamma}[t])$ moves along $\mathbf{1}^{\perp}$ approaching $\sigma(\hat{\gamma}^{\star})$. (An illustrative example of $L = M = 2$ is depicted in Fig. 4.2.) What matters is the distance $||\sigma(\hat{\gamma}[t]) - \sigma(\hat{\gamma}^{\star})||$ restricted on $\mathbf{1}^{\perp}$, which is only affected by the subgradient move $h[t]\tilde{\nabla}\mathcal{L}_{\lambda[t]}(\hat{\gamma}[t])$. Now we have

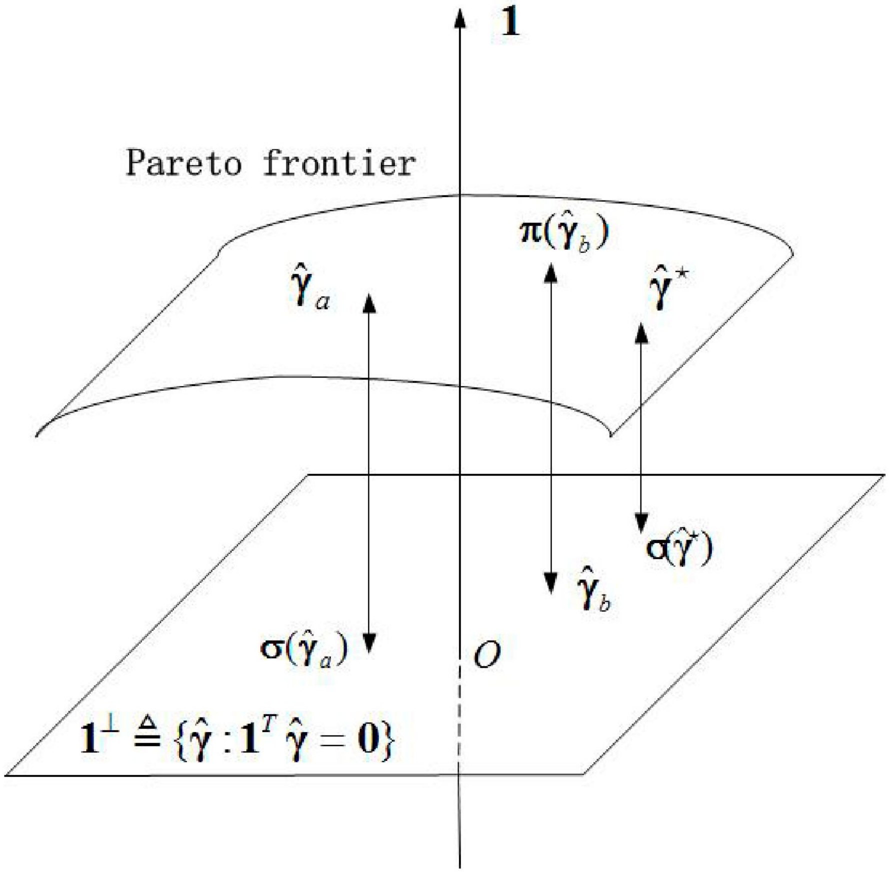

Figure 4.1: Bijective projection mappings between Pareto frontier and $1^\perp$.

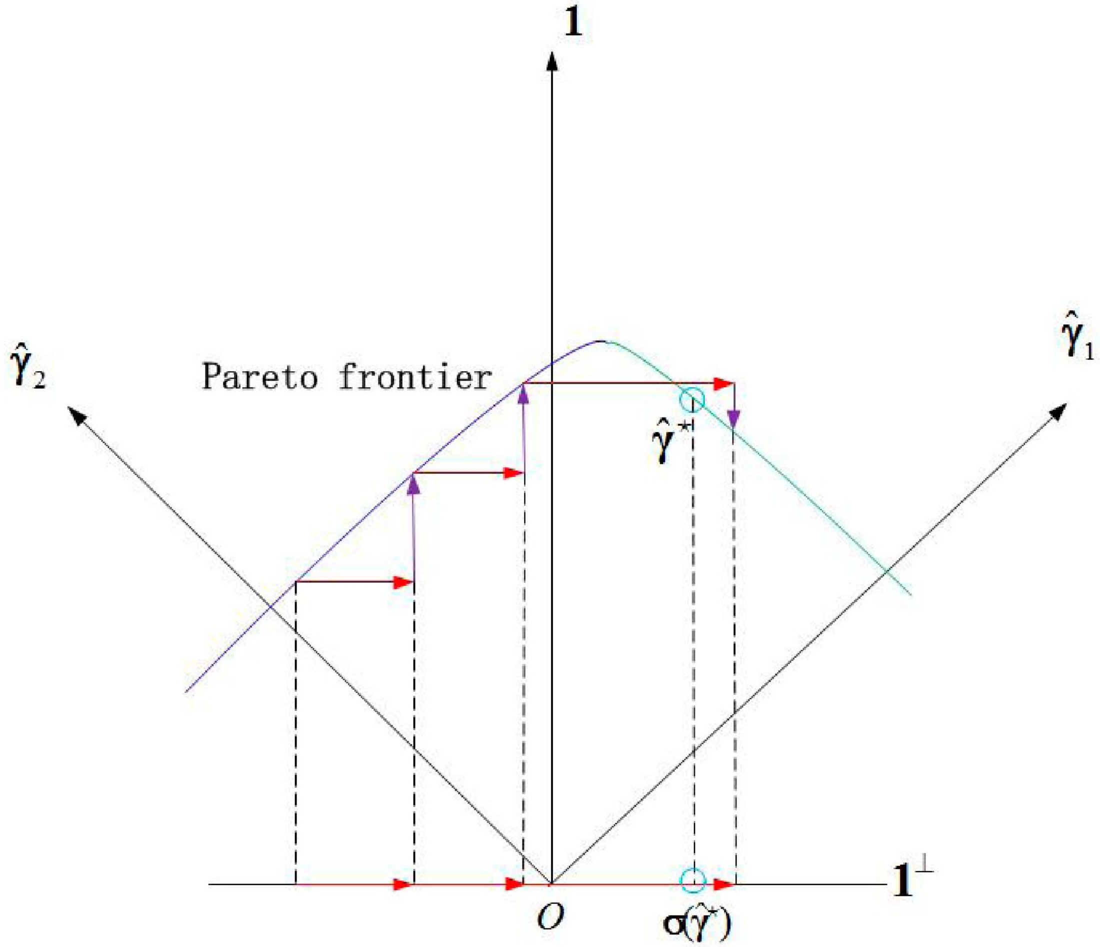

Figure 4.2: An illustrative example of the iterations in Algorithm 4.1 with $L = 2$ links and $M = 2$ constraints. The horizontal (vertical) red (purple) arrows denote the subgradient (projection) moves. The blue and green curves represent respectively two active constraints.

$$
\begin{aligned}
& ||\sigma(\hat{\gamma}[t+1]) - \sigma(\hat{\gamma}^\star)||^2 \\
= \; & ||h[t]\tilde{\nabla}\mathcal{L}_{\lambda[t]}(\hat{\gamma}[t]) + \sigma(\hat{\gamma}[t]) - \sigma(\hat{\gamma}^\star)||^2 \\
= \; & ||\sigma(\hat{\gamma}[t]) - \sigma(\hat{\gamma}^\star)||^2 + h[t]^2||\tilde{\nabla}\mathcal{L}_{\lambda[t]}(\hat{\gamma}[t])||^2 \\
& +2h[t]\tilde{\nabla}\mathcal{L}_{\lambda[t]}(\hat{\gamma}[t])^T[\mathbf{I} - \mathbf{1}\mathbf{1}^T/L]\{\hat{\gamma}[t] - \hat{\gamma}^\star\} \\
= \; & ||\sigma(\hat{\gamma}[t]) - \sigma(\hat{\gamma}^\star)||^2 + h[t]^2||\tilde{\nabla}\mathcal{L}_{\lambda[t]}(\hat{\gamma}[t])||^2 + 2h[t]\tilde{\nabla}\mathcal{L}_{\lambda[t]}(\hat{\gamma}[t])^T\{\hat{\gamma}[t] - \hat{\gamma}^\star\} \\
\leq \; & ||\sigma(\hat{\gamma}[t]) - \sigma(\hat{\gamma}^\star)||^2 + h[t]^2||\tilde{\nabla}\mathcal{L}_{\lambda[t]}(\hat{\gamma}[t])||^2 + 2h[t]\{\mathcal{L}_{\lambda[t]}(\hat{\gamma}[t]) - \mathcal{L}_{\lambda[t]}(\hat{\gamma}^\star)\} \\
= \; & ||\sigma(\hat{\gamma}[t]) - \sigma(\hat{\gamma}^\star)||^2 + h[t]^2||\tilde{\nabla}\mathcal{L}_{\lambda[t]}(\hat{\gamma}[t])||^2 + 2h[t]\{\hat{U}(\hat{\gamma}[t]) - \hat{U}(\hat{\gamma}^\star)\}, \quad \textbf{(4.4)}
\end{aligned}
$$

where the last inequality follows from the concavity of $\mathcal{L}_{\lambda[t]}(\hat{\gamma})$ in $\hat{\gamma}$. This is exactly the well-known basic inequality in the convergence proof and complexity bound analysis of subgradient algorithms. It suggests that our algorithm 4.1 is equivalent to a standard subgradient algorithm on $\mathbf{1}^\perp$. Then follows from the standard treatment and classical results in [22, 23, 24], we have

Lemma 10. *Under the mild assumption that the first order derivatives of* $\hat{U}$ is bounded:

$$\mathbf{1}^T\nabla\hat{U}(\hat{\gamma}) \leq V_\lambda, \forall\hat{\gamma},$$

algorithm 4.1 converges: $\limsup_{t\to\infty}\hat{U}(\hat{\gamma}[t]) = \hat{U}(\hat{\gamma}^\star)$*, with the proper step size rules satisfying* $\sum_{t=0}^{\infty}h[t] = \infty, h[t] \to 0.$

Here is the sketch of the proof. Note that $||\tilde{\nabla}\mathcal{L}_{\lambda[t]}(\hat{\gamma}[t])|| \leq ||\tilde{\nabla}\mathcal{L}_{\lambda[t]}(\hat{\gamma}[t])||_1 \leq$

$2\{1^T\nabla\hat{U}(\hat{\gamma}[t])\} \leq 2V_\lambda$. Then $||\tilde{\nabla}\mathcal{L}_{\lambda[t]}(\hat{\gamma}[t])||$ is uniformly bounded. From (4.4) we have

$$\hat{U}(\hat{\gamma}^\star) - \max_{0\leq t\leq T} \hat{U}(\hat{\gamma}[t]) \leq \frac{||\sigma(\hat{\gamma}[0]) - \sigma(\hat{\gamma}^\star)||^2 + 4V_\lambda^2 \sum_{t=0}^T h[t]^2}{2\sum_{t=0}^T h[t]}.$$

Then simply let $T \to \infty$ concludes the proof.

Note that universally applicable step size rules such as $h[t] = h/(t+1)$ and $h[t] = h/\sqrt{t+1}$ ensure convergence. Note also that the subgradient method and Lemma 10 can be further generalized to the case where $\hat{U}_l(\hat{\gamma}_l)$'s are nondifferentiable, just to replace the mild condition by

$$1^T\tilde{\nabla}\hat{U}(\hat{\gamma}) \leq V_\lambda,\ \forall\tilde{\nabla}\hat{U}(\hat{\gamma}) \in \partial\hat{U}(\hat{\gamma}),\ \forall\hat{\gamma}.$$

The price to pay for universally applicable step size rules, nondifferentiability, and milder condition is that the convergence is in a weaker sense and may be more slower. Convergence to a δ-suboptimal solution $\hat{\gamma}[t]$: $\max_{0\leq t\leq T}\hat{U}(\hat{\gamma}[t]) \geq \hat{U}(\hat{\gamma}^\star) - \delta$ requires $O(\delta^{-2})$ iterations. See, for example, [14, 22] for more details.

4.3 Unconstrained Equivalence and Complexity results of $M = 1$

As a summary to the last two sections, let us provide an alternative perspective to look at our core problem (4.1). We have shown that its optimal solution can only be attained on the Pareto frontier, resulting in the following equivalent formulation:

$$\begin{aligned} \max \quad & \hat{U}(\hat{\gamma}) \\ \text{s.t.} \quad & \hat{\rho}(\hat{\gamma}) = 0. \end{aligned}$$

The bijective mapping between the Pareto frontier $\{\hat{\gamma} : \hat{\rho}(\hat{\gamma}) = 0\}$ and $\mathbf{1}^{\perp} \triangleq \{\hat{\gamma} : \mathbf{1}^T \hat{\gamma} = 0\}$ further transforms the problem to

$$\begin{aligned} \max \quad & \breve{U}(\breve{\gamma}) \triangleq \hat{U}(\pi(\breve{\gamma})) \\ \text{s.t.} \quad & \breve{\gamma} \in \mathbf{1}^{\perp}. \end{aligned}$$

Therefore, the constrained convex optimization problem (4.1) is equivalent to an *unconstrained* convex optimization problem $\max \breve{U}(\breve{\gamma})$ on domain $\mathbf{1}^{\perp}$. (The concavity of $\breve{U}(\breve{\gamma})$ can be readily seen by noting that

$$\hat{U}(\pi(\breve{\gamma})) = \hat{U}(\breve{\gamma} - \mathbf{1}\hat{\rho}(\breve{\gamma})) = \sum_{l \in \langle L \rangle} \hat{U}_l(\breve{\gamma}_l - \hat{\rho}(\breve{\gamma})),$$

in which $\hat{U}_l(\breve{\gamma}_l - \hat{\rho}(\breve{\gamma}))$ is a composition of two concave functions, and invoking the composition rules (3.11) in [19].)

This perspective is useful to resolve the outstanding complexity issue of $M = 1$ near the end of section 4.1, because we can readily adapt the proof of (4.3) in Appendix H to show that $\breve{U}(\breve{\gamma})$ has Lipschitz continuous gradient with the constant κ:

$$\left\|\nabla\breve{U}(\breve{\gamma}) - \nabla\breve{U}(\breve{\beta})\right\| \leq \kappa \left\|\breve{\gamma} - \breve{\beta}\right\|, \forall\breve{\beta},\, \breve{\gamma} \in \mathbf{1}^{\perp}, \tag{4.5}$$

which is the principal premise for the well-known $O(\delta^{-1})$ and $O(\delta^{-1/2})$ complexity bounds given in, e.g., propositions 6.9.1 and 6.9.2 of [24].

Now let us revisit the case where $M = 1$, and $\hat{U}_l(\hat{\gamma}_l)$'s are twice continuously differentiable. Then $\hat{U}(\pi(\breve{\gamma}))$ is also twice continuously differentiable. The steepest ascent direction of such unconstrained problem is given by

$$\begin{aligned}
\nabla\breve{U}(\breve{\gamma}) = \nabla_{\breve{\gamma}}\hat{U}(\pi(\breve{\gamma})) &= \left[\frac{\partial\pi(\breve{\gamma})}{\partial\breve{\gamma}}\right]^T \nabla_{\pi}\hat{U}(\pi(\breve{\gamma})) \\
&= [\mathbf{I} - \nabla_{\breve{\gamma}}\hat{\rho}(\breve{\gamma})\mathbf{1}^T]\nabla_{\pi}\hat{U}(\pi(\breve{\gamma})) \\
&= \nabla_{\pi}\hat{U}(\pi(\breve{\gamma})) - \nabla_{\breve{\gamma}}\hat{\rho}(\breve{\gamma})\mathbf{1}^T\nabla_{\pi}\hat{U}(\pi(\breve{\gamma})) \\
&= \nabla_{\pi}\hat{U}(\pi(\breve{\gamma})) - \nabla_{\breve{\gamma}}\hat{\rho}(\pi(\breve{\gamma}))[\mathbf{1}^T\nabla_{\pi}\hat{U}(\pi(\breve{\gamma}))],
\end{aligned}$$

where $\nabla_{\breve{\gamma}}\hat{\rho}(\breve{\gamma}) = \nabla_{\breve{\gamma}}\hat{\rho}(\pi(\breve{\gamma}))$ follows from the fact that $\nabla_{\breve{\gamma}}\hat{\rho}(\breve{\gamma}+a\mathbf{1}) = \nabla_{\breve{\gamma}}[\hat{\rho}(\breve{\gamma})+a] = \nabla_{\breve{\gamma}}\hat{\rho}(\breve{\gamma})$. We notice that $\nabla_{\breve{\gamma}}\hat{U}(\pi(\breve{\gamma}))$ is exactly $\nabla\mathcal{L}_{\lambda}(\hat{\gamma})$ where $\lambda = \mathbf{1}^T\nabla\hat{U}(\hat{\gamma})$ and $\hat{\gamma} = \pi(\breve{\gamma})$. This manifests the equivalence between our gradient-projection algorithm 4.1 w.r.t. iterate series $\{\hat{\gamma}[t]\}$ and its steepest-ascent-gradient counterpart w.r.t. $\{\breve{\gamma}[t] = \sigma(\hat{\gamma}[t])\}$ given below.

Algorithm 4.2 (Sub)gradient Network Utility Maximization on $1^{\perp}$

1. Initialization. Input an arbitrary $\breve{\gamma}[0] \in \mathbf{1}^{\perp}$ (say, $\breve{\gamma}[0] \leftarrow 0$).
2. Iteration. In $[t+1]$th iteration, set a gradient step $\Delta\breve{\gamma}[t] \leftarrow h[t]\nabla\breve{U}(\breve{\gamma}[t])$. (Therefore, $\breve{\gamma}[t+1] = \breve{\gamma}[t] + \Delta\breve{\gamma}[t]$.) Repeat.

We also notice that

$$\begin{aligned}\nabla^2_{\breve{\gamma}\breve{\gamma}}\hat{U}(\pi(\breve{\gamma})) &= [\mathbf{I} - \nabla_{\breve{\gamma}}\hat{\rho}(\breve{\gamma})\mathbf{1}^T]\nabla^2_{\pi\pi}\hat{U}(\pi(\breve{\gamma}))[\mathbf{I} - \nabla_{\breve{\gamma}}\hat{\rho}(\breve{\gamma})\mathbf{1}^T]^T \\ &\quad -\nabla^2_{\breve{\gamma}\breve{\gamma}}\hat{\rho}(\breve{\gamma})\mathbf{1}^T\nabla_{\pi}\hat{U}(\pi(\breve{\gamma})).\end{aligned}$$

Similar to the proof of (4.3), it is easy to show that

$$-\nabla^2_{\breve{\gamma}\breve{\gamma}}\hat{\rho}(\breve{\gamma})\mathbf{1}^T\nabla_{\pi}\hat{U}(\pi(\breve{\gamma})) \succeq -V_{\lambda}V_{\rho}\mathbf{I},$$

and

$$\begin{aligned}& \Delta\breve{\gamma}^T[\mathbf{I} - \nabla_{\breve{\gamma}}\hat{\rho}(\breve{\gamma})\mathbf{1}^T]\nabla^2_{\pi\pi}\hat{U}(\pi(\breve{\gamma}))[\mathbf{I} - \nabla_{\breve{\gamma}}\hat{\rho}(\breve{\gamma})\mathbf{1}^T]^T\Delta\breve{\gamma} \\ \geq\ & -V_U\Delta\breve{\gamma}^T[\mathbf{I} - \nabla_{\breve{\gamma}}\hat{\rho}(\breve{\gamma})\mathbf{1}^T][\mathbf{I} - \mathbf{1}\nabla_{\breve{\gamma}}\hat{\rho}(\breve{\gamma})^T]\Delta\breve{\gamma} \\ =\ & -V_U\Delta\breve{\gamma}^T[\mathbf{I} + L\nabla_{\breve{\gamma}}\hat{\rho}(\breve{\gamma})\nabla_{\breve{\gamma}}\hat{\rho}(\breve{\gamma})^T]\Delta\breve{\gamma} \\ \geq\ & -V_U(L+1)\Delta\breve{\gamma}^T\Delta\breve{\gamma},\ \forall\Delta\breve{\gamma}, \breve{\gamma} \in \mathbf{1}^{\perp}.\end{aligned}$$

Therefore

$$\Delta\breve{\gamma}^T \nabla^2_{\breve{\gamma}\breve{\gamma}} \hat{U}(\pi(\breve{\gamma}))\Delta\breve{\gamma} \;\; \geq \;\; -\kappa\Delta\breve{\gamma}^T\Delta\breve{\gamma}, \; \forall\Delta\breve{\gamma}, \breve{\gamma} \in \mathbf{1}^\perp,$$

and hence

$$\breve{U}(\breve{\gamma} + \Delta\breve{\gamma}) - \breve{U}(\breve{\gamma}) - \nabla_{\breve{\gamma}}\hat{U}(\pi(\breve{\gamma}))^T\Delta\breve{\gamma} \geq -\kappa\Delta\breve{\gamma}^T\Delta\breve{\gamma}, \; \forall\Delta\breve{\gamma}, \; \breve{\gamma} \in \mathbf{1}^\perp,$$

which implies (4.5), the key premise to apply propositions 6.9.1 and 6.9.2 in [24]. Both propositions assume a constant step size rule $h[t] = 1/\kappa$. From proposition 6.9.1, we get an iteration bound $t \leq \kappa||\breve{\gamma}[0] - \breve{\gamma}^\star||^2/[2\delta] = O(\delta^{-1})$ for convergence to a δ-suboptimal solution $\breve{\gamma}[t] : \breve{U}(\breve{\gamma}[t]) \geq \hat{U}(\breve{\gamma}^\star) - \delta$. This complexity bound can be further improved to $t \leq \sqrt{2\kappa/\delta}||\breve{\gamma}[0] - \breve{\gamma}^\star|| - 1 = O(\delta^{-1/2})$, due to proposition 6.9.2, if we add in a deliberately chosen extrapolation step. Toward this end, we replace the gradient step $\breve{\gamma}[t+1] \leftarrow \breve{\gamma}[t] + h[t]\nabla\breve{U}(\breve{\gamma}[t])$ in algorithm 4.2 by

$$\begin{aligned} \breve{\beta}[t] \;\; &\leftarrow \;\; \breve{\gamma}[t] + \theta[t](\breve{\gamma}[t] - \breve{\gamma}[t-1]) \\ \breve{\gamma}[t+1] \;\; &\leftarrow \;\; \breve{\beta}[t] + h\nabla\breve{U}(\breve{\beta}[t]), \end{aligned}$$

where $h = 1/\kappa$, $\breve{\gamma}[0] = \breve{\gamma}[-1]$ and

$$\theta[t] = \begin{cases} 0 & t = 0 \\ (t-1)/(t+2) & t = 1, 2, \ldots. \end{cases}$$

Equivalently, our algorithm 4.1 has an accelerated variant as well.

Algorithm 4.3 Accelerated (Sub)gradient Projected Network Utility Maximization

```
1. Initialization. Input an arbitrary Pareto efficient $\hat{\gamma}[0]$, let $\Delta\hat{\gamma}^{'}[-1] \leftarrow 0$ and $h \leftarrow 1/\kappa$.
2. Iteration. In $[t+1]$th iteration,
    (a) First compute $\hat{\beta}[t] \leftarrow \hat{\gamma}[t] + \theta[t](\Delta\hat{\gamma}^{'}[t-1])$, and project it to the Pareto efficient $\pi(\hat{\beta}[t])$.
    (b) Secondly, compute $\lambda[t] \leftarrow \mathbf{1}^T \nabla \hat{U}(\pi(\hat{\beta}[t]))$, and $\tilde{\nabla}\mathcal{L}_{\lambda[t]}(\pi(\hat{\beta}[t])) \leftarrow \nabla \hat{U}(\pi(\hat{\beta}[t])) - \lambda[t]\tilde{\nabla}\hat{\rho}(\pi(\hat{\beta}[t]))$.
    (c) Then set a gradient step $\Delta\hat{\gamma}^{''}[t] \leftarrow h\tilde{\nabla}\mathcal{L}_{\lambda[t]}(\pi(\hat{\beta}[t]))$, compute $\Delta\hat{\gamma}^{'}[t] \leftarrow \theta[t](\Delta\hat{\gamma}^{'}[t-1]) + \Delta\hat{\gamma}^{''}[t]$, and project $\hat{\gamma}[t] + \Delta\hat{\gamma}^{'}[t]$ to the Pareto frontier to get $\hat{\gamma}[t+1]$. Repeat.
```

It is well-known that these accelerated algorithms are optimal in terms of the complexity order. In addition, if there exists a positive κ' such that

$$\left\|\nabla \breve{U}(\breve{\gamma}) - \nabla \breve{U}(\breve{\beta})\right\| \geq \kappa' \left\|\breve{\gamma} - \breve{\beta}\right\|, \forall \breve{\beta},\ \breve{\gamma} \in \mathbf{1}^{\perp},$$

then our accelerated algorithms converges geometrically fast: $t = O(-\log\delta)$. See, e.g., Theorem 2.2.2 in [22]. For more details, we refer interested readers to the the standard treatment and classical results in the online supplementary chapter 6 of [24].

Before concluding this section, it is worth pointing out that the accelerated variant of algorithm 4.1 can be readily adapted to the case of $M > 1$. Extensive simulations suggest that algorithm 4.3 still converges quickly (e.g., see Fig. 4.9 in the next section). However, the above convergence and complexity analysis of algorithm 4.3 is only rigorous for the case of $M = 1$. Except for the insight that $\hat{U}(\pi(\check{\gamma}))$ is *almost* differentiable everywhere, we currently have no idea how to extend the *rigorous* analysis to the case of $M > 1$.

4.4 Simulation Experiments

In the following, we will demonstrate by simulation experiments that our algorithms are efficient and optimal, beating its counterparts in prior work [7] and [25].

4.4.1 Simulation Settings

We first describe the settings for the following simulation experiments. We consider two different scenarios: arbitrary random networks and uplink LTE networks.

1. In the arbitrary random networks, we randomly generate i.i.d. channel gain matrices G, noise power n and power constraints $\mathrm{c}_m^T\mathrm{p} \leq$

$1, \forall m \in \langle M \rangle$ of different network sizes L and numbers of constraints M, according to various distributions. Extensive simulations in such type of networks help demonstrate the robustness and universal applicability of our algorithms.

2. In the uplink LTE networks, we adopt similar settings as in [25]. We assume a standard 19-hexagonal-cell layout with wrap around to avoid edge effect, which is widely accepted in LTE cell planning and performance evaluation. Each cell is further divided into 3 sectors. So there are total 57 base stations. Each sector is served by one directional antenna operating at 2GHz to communicate with mobile users who use omnidirectional antennas. The antenna model, path loss model, shadowing model and noise model are specified in [26]. The total uplink system bandwidth is 10MHz and is evenly divided into 10 channels. At any snapshot, each base station will receive packets simultaneously from 10 mobile users with random locations within the corresponding sector. So there are altogether $L = 57$ active links sharing one channel in the network. Extensive simulations in such type of networks help demonstrate the practical applicability of our algorithms.

In both scenarios, two typical network utility functions are chosen for performance evaluation:

1. Equally weighted proportionally fair SINR allocation:

$$U(\gamma) = \sum_{l \in \langle L \rangle} \log(\gamma_l)/L;$$

2. Proportionally fair Shannon rate allocation:

$$U(\gamma) = \sum_{l \in \langle L \rangle} \log(\log(1 + \gamma_l))/L.$$

Note that while the former satisfies the mild condition in lemma 9 with $V_\lambda = 1$ and $V_U = 0$, the latter does not have a bounded V_λ. But it turns out later that the our algorithms can still converge to a neighborhood of the global optimum very quickly in extensive simulations.

Finally, for performance benchmark, we also compute the global optimal solutions via a centralized cvx solver which is built upon our contributed Matlab function "logPFeig". For comparison of different distributed algorithms, we record their numbers of iterations and time for convergence to within 2% or 10% suboptimality. We performed simulation experiments with MATLAB 7.0 on a Intel Core i7 870 desktop.

4.4.2 Negative results of algorithm 6 in [7]

We first report the negative results of algorithm 6 in [7]. Algorithm 6 targets at a special case of our problem (2.2). Specifically, they consider a sole sum power constraint ($M = 1$) and assumes unit noise powers, i.e., $\mathrm{c} = 1/\bar{P}$ and $\mathrm{n} = 1$. Obviously, this special setting naturally satisfies the network duality premise $\mathrm{n}\mathrm{c}^T = \mathrm{c}\mathrm{n}^T$, and gets rid of the involved issues arising from multiple constraints and the case where some components of c are zeros. However, in our simulation, we observe that oftentimes algorithm 6 diverges in the first few iterations, especially when L is large. One link will quickly grab all power and starves other links, resulting in the worst $-\infty$ utility.

We remark that such unfair allocation is due to the holes in the algorithm 6, of which the convergence proof is also incorrect. Note that algorithm 6 is very similar to our algorithm 4.1. According to equation (57) in [7], it also contains a gradient step plus a projection step:

$$\begin{aligned}\hat{\gamma}[t+1] &= \hat{\gamma}[t] - \mathbf{1}\hat{\rho}(\hat{\gamma}[t]) + \Delta\hat{\gamma}'[t] \\ \Delta\hat{\gamma}'[t] &= h[t]\{\frac{\nabla\hat{U}(\hat{\gamma}[t])}{\lambda[t]} \circ [\nabla\hat{\rho}(\hat{\gamma}[t])]^{-1} - \mathbf{1}\}.\end{aligned}$$

However, it is a projection to the Pareto frontier followed by a gradient. The gradient is along the supporting hyperplane of the feasible region since $\Delta\hat{\gamma}'[t]^T\nabla\hat{\rho}(\hat{\gamma}[t]) = 0$. Therefore, the resultant $\hat{\gamma}[t+1]$ must not be feasible, unless $\hat{\gamma}[t] = \hat{\gamma}^\star$. This disproves the feasibility of $\hat{\gamma}[t+1]$ in their convergence proof. More importantly, although such projected gradient move is a diagonally scaled version of our gradient move, it could be descending. Especially when $\nabla\hat{\rho}(\hat{\gamma}[t])_l$ of some link l is relatively too small, a huge step of steep descent is possible. Such link may drive the system to an extremely unfair state as we observed from the simulation. Fortunately, our algorithms provide a fix to it. For instance, consider the equally weighted pro-

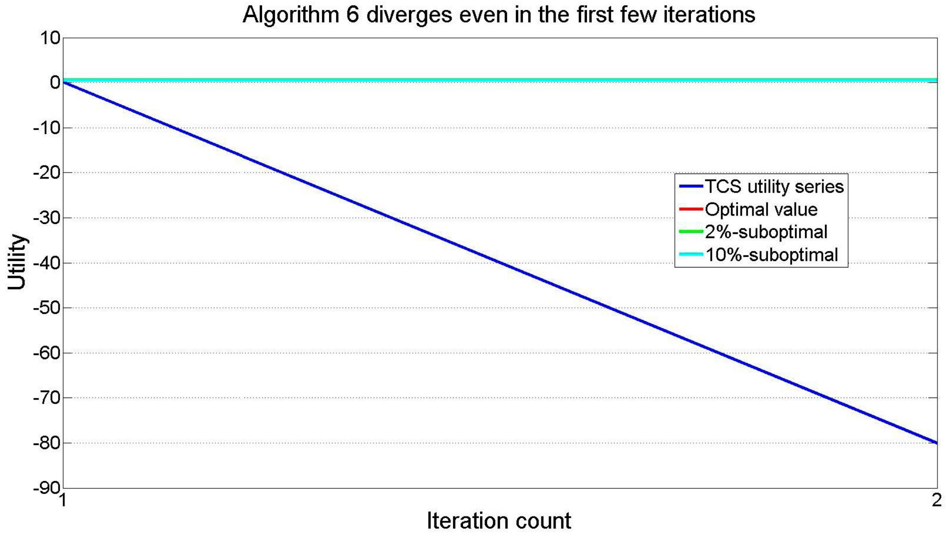

Figure 4.3: Algorithm 6 in [7] diverges. One link quickly grabs all power and starves other links, resulting in the worst $-\infty$ utility just in the first few iterations.

portionally fair SINR allocation problem in an uplink LTE network with one sum power constraint. Starting from the same initial $\hat{\gamma}[0]$, algorithm 6 diverges in the first few iterations, while both our algorithms converge to within-2%-suboptimality in a short time. See figures 4.3, 4.4 and 4.5 for more details. (We remark that the convergence time, say 34ms and 1.5ms in Fig. 4.4, is the ideal per-link computation time, which does not include the time for message passing, distributed gossip algorithms and other overheads of control signaling, similarly hereinafter.)

4.4.3 Negative results of Qualcomm's load-spillage algorithm in [25].

Next we look at the Qualcomm's load-spillage distributed power control algorithm (abbreviated as QLS) in the seminal paper [25]. It also targets at

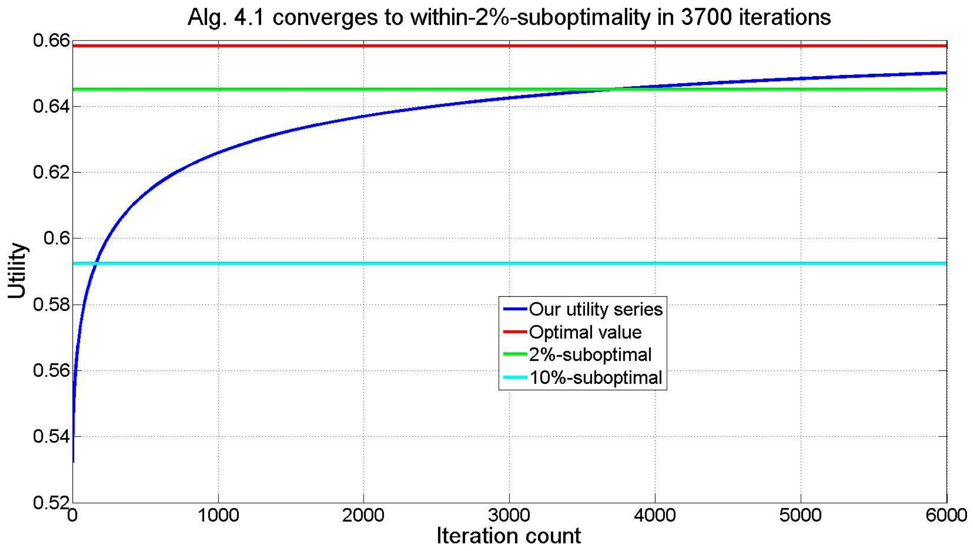

Figure 4.4: Our algorithm 4.1 with diminishing step size rule $h[t] = 1/\sqrt{t+1}$ converges to within-2% (10%)-suboptimality in 3700 (160) iterations or 34ms (1.5ms).

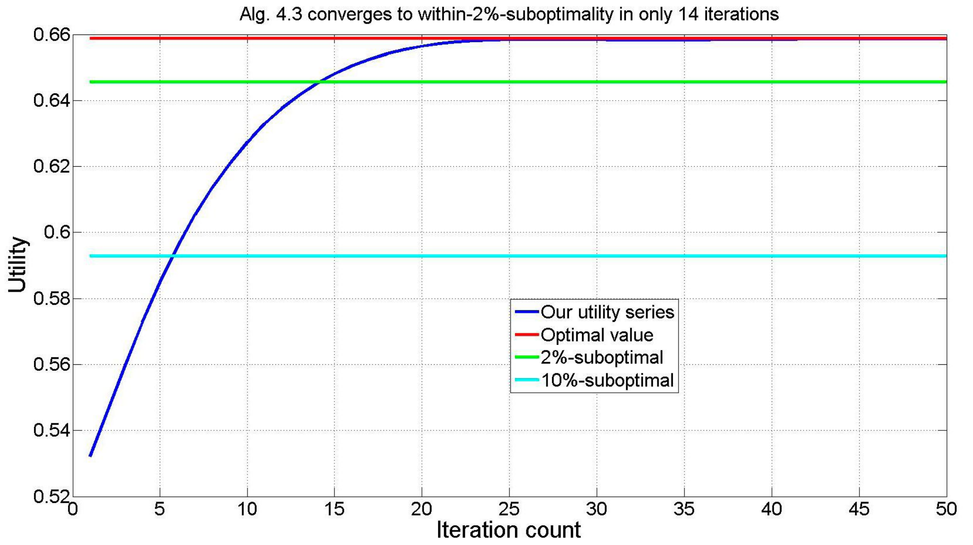

Figure 4.5: Our accelerated algorithm 4.3 with constant step size rule $h[t] = 3.3$ further reduce the convergence time to 14 (6) iterations or 0.33ms (0.16ms).

a special case of our problem (2.2). Specifically, they consider a box power constraint $\mathrm{p} \leq \bar{\mathrm{p}}$ or box interference constraint $\phi \leq \bar{\phi}$. (Obviously, here $M = L$.) They derive a similar network-duality relationship (cf. Fig. 3.1). Roughly speaking, in their paper, “load”, “spillage” and “price” correspond to our dual power, dual interference plus noise and dual noise respectively. Comparison shows that their KKT condition is exactly equivalent to ours (see table 4.1). The difference is that, while we utilize the sparse problem structure and focus on the update of $L+1$ optimizing variables $(\hat{\gamma}, \lambda)$, they try to balance a bunch of variables by distributed pricing. Specifically, at each iteration, they need to update $2L$ optimizing variables (s, ν):

$$\begin{aligned} \mathrm{s}[t+1] &= \mathrm{s}[t] + h_1[t]\Delta\mathrm{s}[t] \\ \nu[t+1] &= \nu[t] + h_2[t](\mathrm{p}[t] - \bar{\mathrm{p}}). \end{aligned}$$

Although they prove that their update corresponds to an ascent direction of utility, the valid step size rule, convergence proof and complexity bound are missing. Only two empirical step size rules are provided in simulation:

1. $h_1[t] = h_1,\ h_2[t] = h_2$;

2. $h_1[t] = h_1,\ h_2[t] = h_2/t$,

where $h_1 = 0.1$ and $h_2 = 0.01$. These rules, however, fail in different settings. For illustration, consider the equally weighted proportionally fair SINR allocation problem in an uplink LTE network with a box power constraint. Starting from the same initial $\hat{\gamma}[0]$, oftentimes QLS either runs

Parameters in [25]	Notations in [25]	Represented in our notations
power	p	$\mathrm{g} \circ \mathrm{p}$
path gain matrix	G	$\bar{\mathrm{G}} diag(\mathrm{g}^{-1})$
interference plus noise	q	ϕ
SINR	γ	γ
load	s	$\tilde{\lambda}\mathrm{p}_d$
spillage	r	$\tilde{\lambda}\mathrm{g}^{-1} \circ \phi_d$
price	ν	$\tilde{\lambda}\mathrm{g}^{-1} \circ \mathrm{n}_d$
KKT condition	$\nabla \hat{U}(\hat{\gamma}^\star) = \mathrm{s}^\star \circ \mathrm{q}^\star$	$\nabla \hat{U}(\hat{\gamma}^\star) = \tilde{\lambda}^\star \mathrm{p}_d^\star \circ \phi^\star$

Table 4.1: Notation correspondence. (At optimality, $\tilde{\lambda}^\star = \lambda^\star / (\mathrm{p}_d^{\star T} \phi^\star)$.)

into non-convergent oscillation (Fig. 4.6) or exhibits extremely slow convergence (Fig. 4.7). Such performance deterioration is mainly due to

1. No valid step size rule: ascent direction by itself can not ensure performance improvement;

2. Foschini-Miljanic distributed power control plus soft pricing $\nu[t+1] = \nu[t] + h_2[t](\mathrm{p}[t] - \bar{\mathrm{p}})$ can not guarantee feasible operating points that obey power constraints;

3. There are too many equations to balance, which is quite sensitive to initialization and the choice of step size rule.

Again, our algorithms also provide a fix to it. Both our algorithms converge to within-2%-suboptimality very quickly. See figures 4.8 and 4.9 for more details.

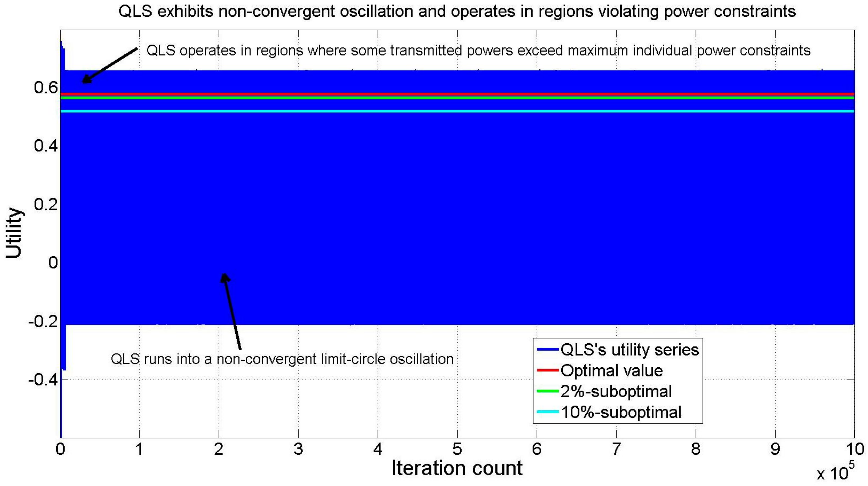

Figure 4.6: QLS with step-size rule 1 does not converge. The overall simulation time is 3.5s or 1 million iterations.

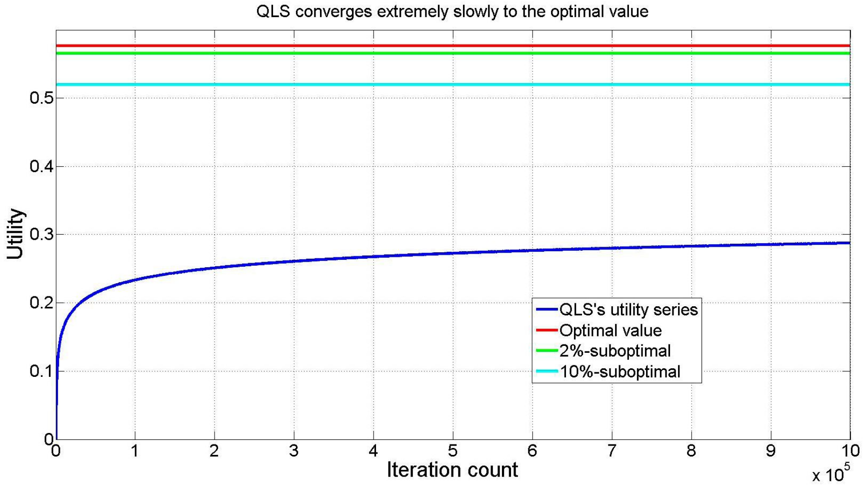

Figure 4.7: QLS with step-size rule 2 converges extremely slowly. The overall simulation time is 5.6s or 1 million iterations.

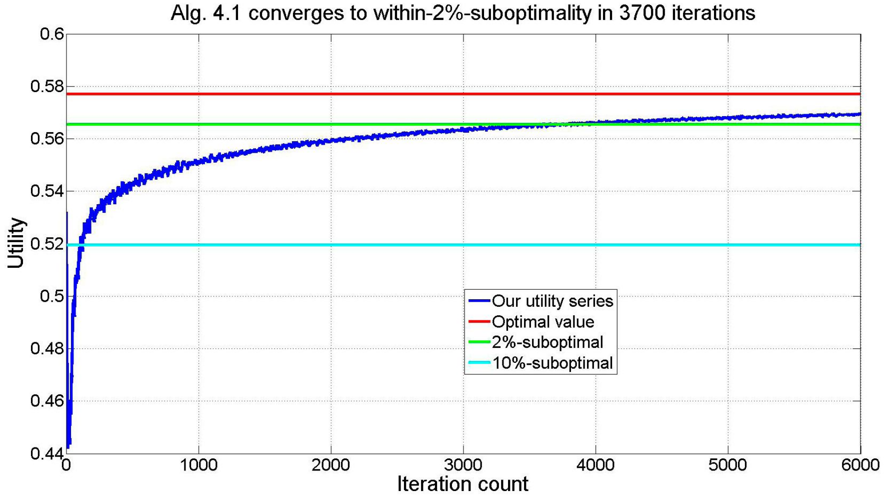

Figure 4.8: Our algorithm 4.1 with diminishing step size rule $h[t] = 1/\sqrt{t+1}$ converges to within-2% (10%)-suboptimality in 3700 (160) iterations or 37ms (1.6ms).

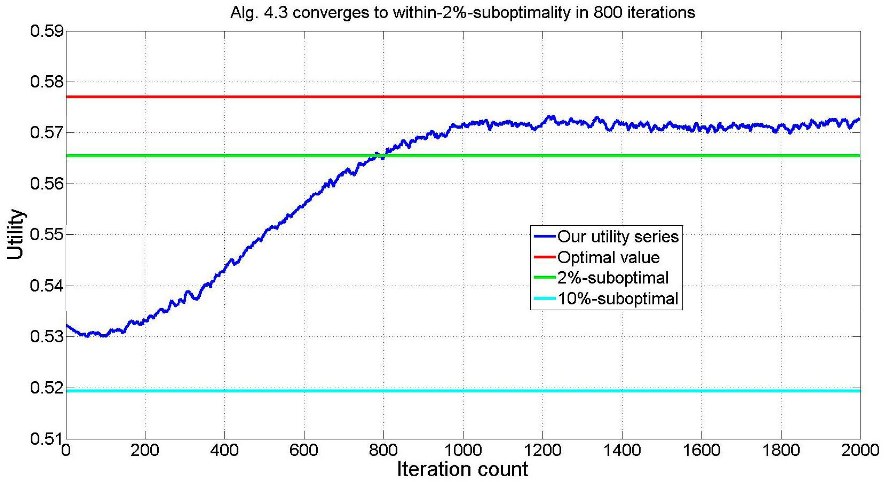

Figure 4.9: Our accelerated algorithm 4.3 with constant step size rule $h[t] = 0.0014$ further reduce the convergence time to 800 iterations or 15ms for within-2% suboptimality.

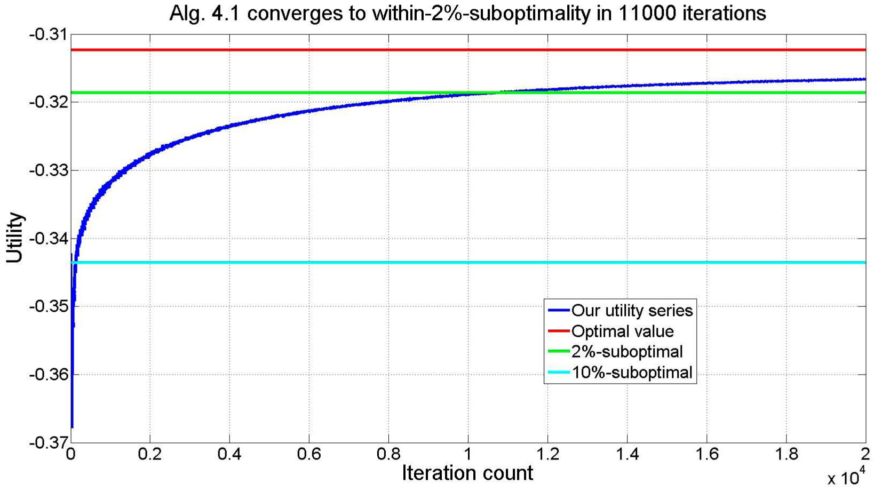

Figure 4.10: For proportionally fair Shannon rate allocation, our algorithm 4.1 with diminishing step size rule $h[t] = 1/\sqrt{t+1}$ converges to within-2% (10%)-suboptimality in 11000 (140) iterations or 110ms (1.4ms).

4.4.4 More results of our algorithms

Our algorithms still preserve good convergence performance in simulation, even if the network utility function is changed to the proportionally fair Shannon rate allocation. An example is shown in Fig. 4.10 (where other settings coincide with those in Fig. 4.8). Note that such utility no longer has a bounded V_λ. Surprisingly, our algorithms can still converge to the neighborhood of the global optimum very quickly.

Consider yet another scenario where an arbitrary random network of $M = L = 10$ random power constraints is assumed. Likewise, our algorithm converges pretty fast. See Fig. 4.11.

We have also performed extensive simulation experiments with other parameter settings. Specifically, we also try various scenarios with different utility functions, different node distributions and topologies, different num-

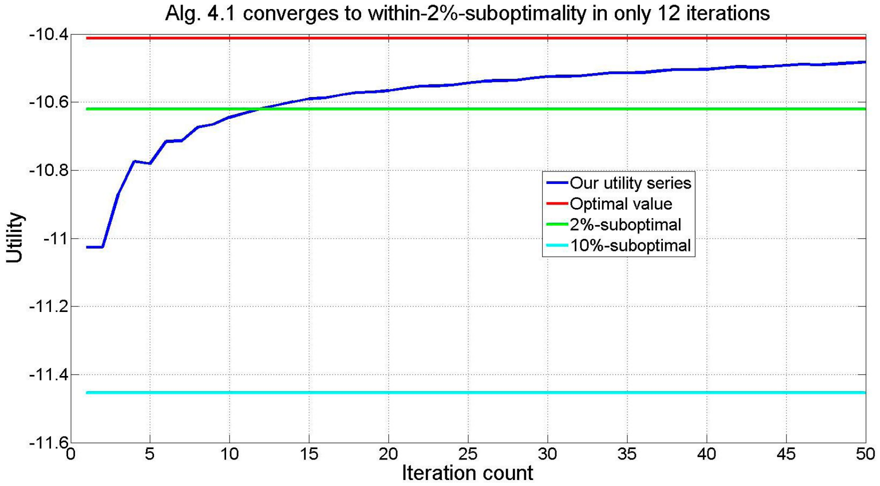

Figure 4.11: For an arbitrary random network of $M = L = 10$ random power constraints, our algorithm 4.1 with diminishing step size rule $h[t] = 1/\sqrt{t+1}$ converges to within-2%-suboptimality in only 12 iterations or 0.83ms.

bers of links L, different numbers of power constraints M. Interestingly, similar results are observed.

To conclude this chapter, we see from both theoretical analysis and simulation experiment that our distributed algorithms are fast and optimal, fixing a number of defects and shortcomings of prior algorithms.

Chapter 5

Related Work

In the last section of chapter 4, we have intensively discussed the connection and difference between our work and the prior work [7] and [25]. Here we briefly summarize them in Table 5.1.

Besides the prior work [7] and [25], there was an even earlier work [31] on this topic using the idea of fictitious game theory. Its Asynchronous Distributed Pricing (ADP) algorithm ensures global fast convergence to the unique optimal solution without using any step size, under a more restricted family of utility functions and box power constraint. (Link-wise utility function $U_l(\gamma_l)$ with $-\frac{\gamma_l U_l''(\gamma_l)}{U_l'(\gamma_l)} \in [1, 2]$ is assumed in [31]. In contrast, $-\frac{\gamma_l U_l''(\gamma_l)}{U_l'(\gamma_l)} \in [1, \infty)$ is assumed in [7], [25] and this thesis. See section 2.3.) Besides the restricted problem scope, another drawback of this algorithm is that it relies on an expensive flooding algorithm to exchange the price globally and periodically; while our algorithms exchange information more

Performance index	Ours	QLS in [25]	Alg. 6 in [7]
Convergence	Guaranteed	Not guaranteed	Not guaranteed
Speed of Convergence	Fast	Slow (if converges)	Initialization dependent
Proof of Convergence	Standard proof in textbook	Sketch of proof. Important details missing.	Algorithm contains defects. Incorrect Proof.
Complexity Bound	Possible and given	Missing	Missing
Power Regulation	Always satisfied and is Pareto optimal	May be violated significantly	May be violated significantly
Sensitivity to Initialization?	Robust to random initialization	Very sensitive, wrong choice results in performance degradation.	Very sensitive, wrong choice results in performance degradation.
Valid Step Size Rule?	Prefix, universally applicable	Missing	Missing
Applicable Scenarios	General nonnegative linear power Constraints	Box power or interference constraint	Sum power constraint

Table 5.1: Comparison between algorithms in prior work and ours

efficiently, thanks to the underlying gossip algorithm (cf. [16]). Nevertheless, there are two advantages of the ADP algorithm over ours. One is that it is provably workable for asynchronous networks. The other is that it requires fewer iterations for convergence. For instance, consider the simulation settings in subsection 4.4.3, the ADP algorithm usually converges to within-2%-suboptimality in less than 10 iterations.

We also notice that the authors of [28] and us independently discovered the solution to the multi-constrained weighted SINR balancing problem (cf. Section 3.3) with different approaches. The subtle difference, roughly speaking, is that we assume a more general **nonnegative** power constraints while they assume **positive** ones. As is discussed in Appendix G, our case is more involved. We have to spend significantly more effort to adapt the underlying previous results from norm to seminorm. More importantly, different from [28], our ultimate goal is general utility maximization, rather than simply balancing the weighted SINR. The multi-constrained weighted SINR balancing only serves as a building block in our network utility maximization algorithms.

Tan, the first author of [7], also generated fruitful research outputs closely related to [7], e.g., [4, 5, 6, 8, 29], just to name a few. We believe that our remedy to [7] strengthen these related works by tying up some of their loose ends.

Finally, we remark that there have been extensive studies on classical distributed or centralized power control problems. We refer interested readers to, e.g., the monograph [20] and the citations therein. In addition, there have been a number of relevant papers on the characterization of the SINR region. Interested readers are referred to [32] and the reference therein.

Chapter 6

Conclusion

THE main contributions of this thesis can be summarized as follows:

1. The investigations here are a first attempt to uncover and fix some of the shortcomings of prior distributed network utility maximization algorithms in [7] and [25]. Observation shows that oftentimes the former diverges in the first few iterations; and the latter fails to converge or converges extremely slowly. Careful examination reveals some of their drawbacks and suggests a pertinent remedy through reformulation of the optimization approach to attack the problem.

2. A novel distributed (sub)gradient projection power control algorithm 4.1 has been devised to fill the gap, based on rigorous derivation. First, we develop key theorem to transform a linear power constraint in power domain to a convex differentiable log-Perron-Frobenius constraint in the log-SINR domain, making our core problem to be a standard convex constrained optimization problem. Second, we show that

the derivative of such log-Perron-Frobenius constraint can also be efficiently distributedly evaluated at any given log-SINR. At the end of such computation, the log-SINR will automatically be projected to the Pareto frontier, ensuring the iterates' Pareto efficiency.

3. An elegant bijective mapping between Pareto frontier and the $1^\perp$ hyperplane has been discovered. The mapping simply transforms the above convex constrained optimization problem to an equivalent unconstrained counterpart, and our (sub)gradient projection algorithm to an equivalent standard (sub)gradient algorithm on $1^\perp$.

4. Backed by rich theory of (sub)gradient algorithms, off-the-shelf convergence proof and complexity analysis are readily applicable. Universally applicable diminishing step size for subgradient algorithm 4.1 and properly chosen constant step size for accelerated (sub)gradient algorithm 4.3 are proposed. The new algorithms easily outperform the algorithms of prior work in terms of efficiency and optimality in simulation experiments.

5. Last but not least, the applicability of our analytical and algorithmic framework is wider in scope than prior work [7] and [25]. It can take care of multiple general nonnegative linear power constraints, which are critical in the control and optimization of today's LTE and cognitive radio networks.

In addition to the above contributions, we believe that our novel algorithms and the corresponding analysis framework provided in this thesis open up many fruitful areas for further research. Some examples are given below:

Asynchronous Power Update – This thesis has implicitly assumed synchronous power update. It is well known that Foschini-Miljanic algorithm and the ADP algorithm in [31]work well also in the asynchronous case. A generalization to the asynchronous case, say, for the multi-constrained weighted SINR balancing problem, will be interesting.

Nonlinear Power Constraint – This thesis assumes general linear power constraints. In general, power constraints can be nonlinear [30]. Is the Perron Frobenius transformation from power domain to SINR domain is still valid for nonlinear power constraints? What will be the image of an nonlinear power constraint?

Math Properties of our (Sub)gradient "Projection" Method – Note that our (sub)gradient "projection" method is a new member to the family of generalized subgradient projection methods [22, 23, 24]. We are interested to know its "designated use" as well as its connection and difference with other "family members", especially those popular ones such as approximate (sub)gradient method, incremental (sub)gradient method and proximal method.

Nonconvex Optimization – This thesis limits its scope to convex optimization. If nonconvex utility is considered, the good properties in the proof of Lemma 8 will be gone. For example, KKT condition may no longer be sufficient for optimality. We may end up with a local optimum. Can we still have some performance guarantee?

We believe the above will be promising directions for our future research.

Chapter 7

Appendix

A. Proof of Lemma 1

Proof. Since $\Gamma_c \subset \Gamma$, we have the Neumann expansion $\mathrm{p}(\gamma) = \sum_{k=0}^{\infty}[diag(\gamma \circ \mathrm{g}^{-1})\bar{\mathrm{G}}]^k diag(\gamma \circ \mathrm{g}^{-1})\mathrm{n}$ from equation (2.1). It implies (2.3) and the equality case. Further, since $\mathrm{c} \gneqq 0$, if $\gamma \in \Gamma_c$ and $\gamma \geq \beta \geq 0$, we have $\mathrm{p}(\gamma) \geq \mathrm{p}(\beta) \geq 0$, and hence $1 \geq \mathrm{c}^T\mathrm{p}(\gamma) \geq \mathrm{c}^T\mathrm{p}(\beta)$. Therefore $\beta \in \Gamma_c$, this shows the monotonicity of the set Γ_c. □

Remark: this lemma is an extension and a direct consequence of Lemma 2.2 in [8], which just simply replaces all Γ_c's with Γ's in our Lemma 1.

B. Proof and Remark of Lemma 2

Proof. Substituting $c^T p = 1$ into $\lambda p = Ap + b$ gives $\lambda p = Ap + bc^T p = (A + bc^T)p$, which is a classical nonnegative-matrix eigenvalue problem. Note that the irreducibility of A carries over to $A + bc^T$. (Nonnegative square matrix A is irreducible if and only if $(\mathbf{I} + A)^{L-1} > 0$ or $\sum_{l=0}^{L-1} A^l > 0$ [10].) The well-known Perron-Frobenius theorem ensures that nonnegative irreducible matrix $A + bc^T$ has only one positive eigenvector p_* (Perron vector), associated with the unique Perron-Frobenius eigenvalue $\lambda_* = \rho(A + bc^T) > 0$. Moreover, p_* is positive and uniquely determined by $c^T p_* = 1$. □

Remark: Notice that previous work [8] and [7] shows $\rho(diag(\gamma \circ g^{-1})(\bar{G} + nc^T)) \leq 1$ for the special case $c = e_l/\bar{p}_l$ and $c = 1/\bar{P}$ respectively. The former uses determinant analysis, while the latter is based on Theorem 6 and Lemma 8 in [1]. The former method is difficult to generalize, while the latter one can be generalized to the case where $c > 0$. In fact, Theorem 6 and Lemma 8 in [1] requires $c^T p$ to be a norm of p so that the corresponding dual norm exists. If $c > 0$, $c^T p = \|p\| := \sum_{l \in \langle L \rangle} c_l |p_l|$ is a norm of p, with dual norm $\|y\|^D := max_l \frac{|y_l|}{c_l}$. However, if c contains some zero components, $c^T p$ is a seminorm without dual seminorm. The results of [1] thus fail to apply. Our Lemma 2, albeit not as elegant as Theorem 6 in [1], is tailored to the unified proof of $c \gneq 0$.

C. Characterization of Γ_c: The Case Where Some Components of γ Are 0

For the case where some components of γ are 0, we notice that $p_l = 0 \Leftrightarrow \gamma_l = 0$, which means some links are not transmitting. Let $\mathcal{A}$ denote the index set of the active transmitting links.We can simply repeat the previous investigation by restricting consideration on $\mathcal{A}$. Denote $\gamma(\mathcal{A})$ the vector composed of positive entries of γ, and $\mathrm{B}(\mathcal{A})$ the principal submatrix of B with the rows and columns in $\mathcal{A}$. (It is easy to see that $\mathrm{B}(\mathcal{A}) = diag(\mathrm{g}(\mathcal{A})^{-1})(\bar{\mathrm{G}}(\mathcal{A}) + \mathrm{n}(\mathcal{A})\mathrm{c}(\mathcal{A})^T)$.) If $\mathrm{c}(\mathcal{A}) \neq 0$, or equivalently, $\mathrm{c}(\mathcal{A})^T\mathrm{p}(\mathcal{A}) > 0$, by the same token, we still have Pareto frontier $\{\gamma(\mathcal{A}) : \rho(diag(\gamma(\mathcal{A}))\mathrm{B}(\mathcal{A})) = 1\}$ and feasible SINR region $\{\gamma(\mathcal{A}) : \rho(diag(\gamma(\mathcal{A}))\mathrm{B}(\mathcal{A})) \leq 1\}$. It is straightforward to see

$$\rho(diag(\gamma)\mathrm{B}) = \rho(diag(\gamma(\mathcal{A}))\mathrm{B}(\mathcal{A})),$$

so the above conclusion remains unchanged. The rest is the “strange” case where $\mathrm{c}(\mathcal{A}) = 0$. In this case, proportionally increasing the power p will no longer affect the constraint $\mathrm{c}(\mathcal{A})^T\mathrm{p}(\mathcal{A})$, which remains 0. In fact, this is similar to the scenario with no power constraint. Then we have $\rho(diag(\gamma)\mathrm{B}) = \rho(diag(\gamma(\mathcal{A}))\mathrm{B}(\mathcal{A})) = \rho(diag(\gamma(\mathcal{A}) \circ \mathrm{g}(\mathcal{A})^{-1})\bar{\mathrm{G}}(\mathcal{A})) < 1$. For such $\gamma(\mathcal{A})$, one can always activate a link $l \notin \mathcal{A}$ with positive c_l by setting its power to be $p_l = 1/c_l$. At the same time, all links in $\mathcal{A}$ proportionally increase their powers by a factor of $\max_{j\in\mathcal{A}}(1 + g_{jl}/c_l n_j)$. It can be shown that the resultant SINR $\gamma^{'}(\mathcal{A}\cup\{l\})$ is Pareto optimal and strictly dominates the original $\gamma(\mathcal{A} \cup \{l\})$. Thus, we can still restrict our focus on the Pareto frontier.

D. Proof of Lemma 3

Proof. This lemma directly follows from our Lemma 2 and Lemma 8, 3 and 4 of [1]. For completeness, we restate those lemmas in [1] as follows:

Let A be a nonnegative irreducible matrix, and $\mathrm{b}, \mathrm{c} \gneqq 0$ two nonnegative vectors. Then

a) $\rho(\mathrm{A} + \mathrm{b}\mathrm{c}^T) > \rho(\mathrm{A})$, and the matrix $\mathrm{A} + \mathrm{b}\mathrm{c}^T$ has only one Perron vector;

b) For any nonnegative vector d, if the matrices $\mathrm{A} + \mathrm{b}\mathrm{c}^T$ and $\mathrm{A} + \mathrm{b}\mathrm{d}^T$ have the same spectral radius $\rho(\mathrm{A} + \mathrm{b}\mathrm{c}^T) = \rho(\mathrm{A} + \mathrm{b}\mathrm{d}^T)$, then their normalized Perron vector are equal;

c) Let p denote the normalized Perron vector of $\mathrm{A} + \mathrm{b}\mathrm{c}^T$, then $\rho(\mathrm{A} + \mathrm{b}\mathrm{c}^T) \overset{>}{\underset{<}{=}} \rho(\mathrm{A} + \mathrm{b}\mathrm{d}^T) \Longleftrightarrow \mathrm{c}^T\mathrm{p} \overset{>}{\underset{<}{=}} \mathrm{d}^T\mathrm{p}$. □

E. Details on How We Design "logPFeig.m" in cvx

Note that any cvx-recognizable function must obey the disciplined convex programming ruleset. The idea of our source code is that we transform the function $f_{\mathrm{B}}(\hat{\gamma})$ to be the solution of a partially minimized convex problem. First, note that from Perron-Frobenius theorem, the Perron (right) eigenvector associated with an irreducible non-negative matrix is unique (up to a scaling factor) and positive. Hence, for the irreducible non-negative matrix $diag(e^{\hat{\gamma}})\mathrm{B}$, let its Perron eigenvector be $\mathrm{x} > 0$. One can easily prove that its Perron eigenvalue $\rho(diag(e^{\hat{\gamma}})\mathbf{B})$ is the optimal value of the following optimization problem

$$\begin{aligned} \min \quad & \lambda \\ s.t. \quad & diag(e^{\hat{\gamma}})\mathbf{B}\mathbf{x} \leq \lambda \mathbf{x} \\ & \mathbf{x} > 0 \end{aligned} \tag{7.1}$$

It is easy to see that any feasible λ must be positive. Let $\hat{\lambda} = \log \lambda$, and $\hat{x_i} = \log x_i$ for all $i \in \langle L \rangle$. Then (7.1) is equivalent to the following:

$$\begin{aligned} \min \quad & \hat{\lambda} \\ s.t. \quad & \log(\sum_{j=1}^{L} B_{ij} \exp(\hat{x_j})) \leq \hat{\lambda} + \hat{x_i} - \hat{\gamma_i} \\ & \forall i \in \langle L \rangle \end{aligned} \tag{7.2}$$

Note that the L.H.S. of the constraints in (7.2) are log-sum-exp functions and the R.H.S. linear functions. So (7.2) is a disciplined convex optimization problem. Also, from the "partial minimization rule" (Section 5.2 in [12]), the optimal value of (2), $f_{\mathbf{B}}(\hat{\gamma}) = \log \rho(diag(e^{\hat{\gamma}})\mathbf{B})$ must be a convex function in $\hat{\gamma}$, which gives an alternative proof of its convexity. Finally, the specification of $f_{\mathbf{B}}(\hat{\gamma})$ in this way totally obeys the cvx rules, and hence can be recognized by cvx and readily used.

F. Proof of Lemma 4

Proof. The following derivation is inspired by the one in pp12 of [14]. From the property of spectral radius, we have

$$h(\hat{\gamma}) := \rho(diag(e^{\hat{\gamma}})\mathbf{B}) = \sup_{\mathbf{y}^T\mathbf{x}=1} \mathbf{y}^{\mathbf{T}} diag(e^{\hat{\gamma}})\mathbf{B}\mathbf{x}.$$

So $h(\hat{\gamma})$ is a point-wise supremum of $g_{\mathbf{y},\mathbf{x}}(\hat{\gamma}) := \mathbf{y}^{\mathbf{T}} diag(e^{\hat{\gamma}})\mathbf{B}\mathbf{x}$ over $\mathbf{y}^T\mathbf{x} = 1$. Note that the gradient of $g_{\mathbf{y},\mathbf{x}}$ with respect to $\hat{\gamma}$ is

$$\nabla g_{\mathbf{y},\mathbf{x}}(\hat{\gamma}) = [y_1 \exp(\hat{\gamma}_1)\mathbf{B}_1\mathbf{x}, \ldots, y_L \exp(\hat{\gamma}_L)\mathbf{B}_L\mathbf{x}]^T,$$

where B_l is the lth row of B. Hence, the subdifferential of $h(\hat{\gamma})$ is $\partial h(\hat{\gamma}) = \{\nabla g_{\mathbf{y},\mathbf{x}}(\hat{\gamma}) | diag(e^{\hat{\gamma}})\mathbf{B}\mathbf{x} = \rho(diag(e^{\hat{\gamma}})\mathbf{B})\mathrm{x}, \mathbf{y}^T diag(e^{\hat{\gamma}})\mathbf{B} = \mathbf{y}^T\rho(diag(e^{\hat{\gamma}})\mathbf{B}), \mathbf{y}^T\mathbf{x} = 1\}$. Note that $diag(e^{\hat{\gamma}})\mathrm{B}$ is an irreducible nonnegative matrix, then $\rho(diag(e^{\hat{\gamma}})\mathrm{B})$ is a simple root of its characteristic polynomial, and its right and left eigenvector $\mathrm{x}(diag(e^{\hat{\gamma}})\mathrm{B})$ and $\mathrm{y}(diag(e^{\hat{\gamma}})\mathrm{B})$ are unique up to multiplication by constant from Perron-Frobenius theorem. Hence $\partial h(\hat{\gamma})$ is a single-point set, ,and $h(\hat{\gamma})$ is differentiable. We have

$$\nabla h(\hat{\gamma}) = [y_1 \exp(\hat{\gamma}_1)\mathbf{B}_1\mathbf{x}, \ldots, y_L \exp(\hat{\gamma}_L)\mathbf{B}_L\mathbf{x}]^T = \rho[y_1x_1, \ldots, y_Lx_L]^T = \rho\mathbf{x} \circ \mathbf{y}.$$

(Here we abbreviate $\mathrm{x}(diag(e^{\hat{\gamma}})\mathrm{B})$, $\mathrm{y}(diag(e^{\hat{\gamma}})\mathrm{B})$ and $\rho(diag(e^{\hat{\gamma}})\mathrm{B})$ by x, y and ρ respectively.) Finally, we have

$$\nabla \log h(\hat{\gamma}) = \nabla h(\hat{\gamma})/h(\hat{\gamma}) = \mathbf{x}(diag(e^{\hat{\gamma}})\mathbf{B}) \circ \mathbf{y}(diag(e^{\hat{\gamma}})\mathbf{B}),$$

where $\mathrm{x}(diag(e^{\hat{\gamma}})\mathbf{B})$ and $\mathrm{y}(diag(e^{\hat{\gamma}})\mathbf{B})$ are normalized such that

$$\mathrm{y}(diag(e^{\hat{\gamma}})\mathbf{B})^T\mathrm{x}(diag(e^{\hat{\gamma}})\mathbf{B}) = 1.$$

□

G. Sketch of the Proof and Remark of Lemma 5

Proof. If $\mathrm{c} > 0$, $\mathrm{c}^T\mathrm{p}$ is a norm of p. Then our proof is done by invoking Theorem 1 in [2]. However, if c contains some zero components, $\mathrm{c}^T\mathrm{p}$ is a seminorm rather than a norm. (Note: a seminorm is essentially a norm with the positivity property removed. The positivity property means that $\|\mathrm{p}\| = 0$ only if $\mathrm{p} = 0$.) Then Theorem 1 in [2] is no longer applicable. Fortunately, we find out a way to bypass the positivity property of the norm, after carefully examining its major proof given in [3].

The idea of our proof is similar to that of Theorem 1 in [2]. For Hilbert's projective metric on $\mathbb{R}^L_{++}$ given by

$$d(\mathbf{p}, \mathbf{q}) = -\log[(\min_{i\in\langle L\rangle} p_i/q_i)(\min_{i\in\langle L\rangle} q_i/p_i)],$$

one shows that $X = \{\mathrm{p} \in \mathbb{R}^L_{++}|\mathrm{c}^T\mathrm{p} = 1\}$ is a complete metric space. Then one invokes the only theorem in [3] to show that $\tilde{T}$ is a contraction for d **asymptotically**. Instead of showing that $\tilde{T} : X \to X$ is a contraction for d as adopted in[2], we show that $\tilde{T}$ is a contraction for d **after the first**

iteration. Then one can apply Banach's contraction mapping principle to (X, d) and $\tilde{T}$ to complete the proof.

In the following we only highlight the part to show $\tilde{T}$ is a contraction for d **after the first iteration**. By carefully examining the proof of the only theorem in [3], we note that its one-iteration-delay variant is still valid, as shown below:

Let T be the mapping operator such that $T\mathrm{p} = \mathbf{A}\mathrm{p} + \mathrm{b}$. Then $s(\mathrm{p}) = \mathrm{c}^T\mathrm{p}$ is a scale function, and $\tilde{T}\mathrm{p} = T\mathrm{p}/s(\mathrm{p})$. (A continuous mapping $s : \mathbb{R}_+^L \to \mathbb{R}_+$ is called a scale function if $s(\mathrm{p})$ is not identically 0; $s(\lambda\mathrm{p}) = \lambda s(\mathrm{p})$ for $\mathrm{p} \geq 0, \lambda \geq 0$; and $0 \leq \mathrm{p} \leq \mathrm{q}$ implies $s(\mathrm{p}) \leq s(\mathrm{q})$.) Now suppose the mapping T satisfies the following conditions:

(i) $\exists$ numbers $\alpha, \beta > 0$ (which only depends on the parameter **A**, b and c) such that $\alpha\mathbf{1} \leq T\tilde{T}\mathrm{p} \leq \beta\mathbf{1}, \forall\mathrm{p} \in X$.

(ii) For any $\mathrm{p}, \mathrm{q} \in X$ and $0 \leq \lambda \leq 1$: If $\lambda\mathrm{p} \leq \mathrm{q}$ then $\lambda T\mathrm{p} \leq T\mathrm{q}$ and if $\lambda\tilde{T}\mathrm{p} \leq \tilde{T}\mathrm{q}$ with $\lambda < 1$, then $\lambda T\mathrm{p} < T\mathrm{q}$.

Then T has the following properties:

(a) $T\mathrm{p} = \lambda\mathrm{p}$ has a unique solution $\mathrm{p}_* \in X$, $\lambda_* > 0$.

(b) $\lim_{k\to\infty} \tilde{T}^k\mathrm{p} = \mathrm{p}_*$ for all $\mathrm{p} : s(\mathrm{p}) > 0$.

To show the conditions (i) and (ii) are satisfied, the key step is to show that $\{T\tilde{T}\mathrm{p} : \forall\mathrm{p} \in X\}$ is uniformly bounded above by some positive vector which only depends on the parameter **A**, b and c. Since T is a bounded nonnegative operator, it suffices to show that $\{\tilde{T}\mathrm{p} : \forall\mathrm{p} \in X\}$ is bounded above. Without loss of generality, we assume $c_l > 0, \forall l \in \{1, \ldots, l_0\}$, and $c_l = 0, \forall l \in \{l_0 + 1, \ldots, L\}$. Since $\beta, \mathbf{G}, \mathbf{n} > 0$, we have $\mathbf{b} > 0$, $A_{ij} > 0$ if $i \neq j$ and $A_{ij} = 0$, otherwise. For arbitrary $\mathbf{p} \in X$, let $\mathbf{z} := \tilde{T}\mathbf{p}$, we must have $\mathbf{z} \in X$. Hence from $\mathbf{c}^T\mathbf{p} = 1$ and $\mathbf{c}^T\mathbf{z} = 1$, we have $p_l, z_l \leq 1/c_l, \forall l \in \{1, \ldots, l_0\}$. For any $l \in \{l_0 + 1, \ldots, L\}$, we have

$$z_l = \frac{(\mathbf{Ap} + \mathbf{b})_l}{\mathbf{c}^T(\mathbf{Ap} + \mathbf{b})} \leq \frac{b_l}{\mathbf{c}^T\mathbf{b}} + \sum_{j=1}^{l_0} \frac{A_{lj}}{c_l\mathbf{c}^T\mathbf{b}} + \sum_{j=l_0+1}^{L} \frac{A_{lj}}{c_1 A_{1j}}.$$

Hence $\mathbf{z}$ is uniformly bounded above, so is $T\mathbf{z}$.

On the other hand, since we have $T\mathbf{p} = \mathbf{Ap} + \mathbf{b} \geq \mathbf{b} > 0$, we can easily show that the remainder conditions are also satisfied. The rest of the proof is almost the same as in [2]. To conserve space, we omit it here. □

Remark: Note that if a link sets its target SINR to be 0, its power will remain 0 at every iteration, then it can be regarded as inactive. Using similar argument as discussed in Appendix C that we can restrict consideration on the set of active links $\mathcal{A}$, one can easily show that the above discussion is still valid for $\beta \gneqq 0$, except that the mild condition on initial power becomes $\mathbf{c}(\mathcal{A})^T\mathbf{p}(\mathcal{A}) > 0$. Also, one can easily adapt the proof to the case where $s(\mathbf{p}) = \max_{m \in \langle M \rangle} (\mathbf{c}_m^T\mathbf{p})$.

H. Proof of Equation (4.3)

Proof. From middle value theorem, we have

$$\hat{U}(\hat{\gamma}[t+1]) - \hat{U}(\hat{\gamma}[t]) = \nabla\hat{U}(\hat{\gamma}[t])^T\Delta\hat{\gamma}[t] + \Delta\hat{\gamma}[t]^T\nabla^2\hat{U}(\hat{\gamma}[t] + \alpha_1\Delta\hat{\gamma}[t])\Delta\hat{\gamma}[t]/2,$$

where $0 < \alpha_1 < 1$.

$$\begin{aligned}
& \nabla\hat{U}(\hat{\gamma}[t])^T\Delta\hat{\gamma}[t] \\
= \; & \{\nabla\mathcal{L}_{\lambda[t]}(\hat{\gamma}[t]) + \lambda[t]\nabla\hat{\rho}(\hat{\gamma}[t])\}^T\{h[t]\nabla\mathcal{L}_{\lambda[t]}(\hat{\gamma}[t]) - \mathbf{1}\hat{\rho}(\hat{\gamma}[t] + h[t]\nabla\mathcal{L}_{\lambda[t]}(\hat{\gamma}[t]))\} \\
= \; & h[t]||\nabla\mathcal{L}_{\lambda[t]}(\hat{\gamma}[t])||^2 + \lambda[t]h[t]\nabla\hat{\rho}(\hat{\gamma}[t])^T\nabla\mathcal{L}_{\lambda[t]}(\hat{\gamma}[t]) - \lambda[t]\hat{\rho}(\hat{\gamma}[t] + h[t]\nabla\mathcal{L}_{\lambda[t]}(\hat{\gamma}[t])) \\
= \; & h[t]||\nabla\mathcal{L}_{\lambda[t]}(\hat{\gamma}[t])||^2 - \lambda[t]h[t]^2\nabla\mathcal{L}_{\lambda[t]}(\hat{\gamma}[t])^T\nabla^2\hat{\rho}(\hat{\gamma}[t] + \alpha_2\Delta\hat{\gamma}^{'}[t])\nabla\mathcal{L}_{\lambda[t]}(\hat{\gamma}[t])/2,
\end{aligned}$$

where $0 < \alpha_2 < 1$.

Applying the mild conditions, we have

$$\hat{U}(\hat{\gamma}[t+1]) - \hat{U}(\hat{\gamma}[t]) \geq h[t]||\nabla\mathcal{L}_{\lambda[t]}(\hat{\gamma}[t]))||^2 - V_\lambda V_\rho h[t]^2||\nabla\mathcal{L}_{\lambda[t]}(\hat{\gamma}[t]))||^2/2 - V_U||\Delta\hat{\gamma}[t]||^2/2.$$

Note that

$$||\Delta\hat{\gamma}[t]||^2 = ||\Delta\hat{\gamma}^{'}[t]||^2 + ||\mathbf{1}\hat{\rho}(\hat{\gamma}[t] + \Delta\hat{\gamma}^{'}[t])||^2 = ||\Delta\hat{\gamma}^{'}[t]||^2 + L\hat{\rho}^2(\hat{\gamma}[t] + \Delta\hat{\gamma}^{'}[t]),$$

and

$$\begin{aligned}
& \hat{\rho}^2(\hat{\gamma}[t] + \Delta\hat{\gamma}^{'}[t]) \\
= \;& ||\Delta\hat{\gamma}^{'}[t]^T \nabla\hat{\rho}(\hat{\gamma}[t] + \alpha_3\Delta\hat{\gamma}^{'}[t])||^2 \\
\leq \;& ||\Delta\hat{\gamma}^{'}[t]||^2 ||\nabla\hat{\rho}(\hat{\gamma}[t] + \alpha_3\Delta\hat{\gamma}^{'}[t])||^2 \\
\leq \;& ||\Delta\hat{\gamma}^{'}[t]||^2 \mathbf{1}^T \nabla\hat{\rho}(\hat{\gamma}[t] + \alpha_3\Delta\hat{\gamma}^{'}[t]) \\
= \;& ||\Delta\hat{\gamma}^{'}[t]||^2,
\end{aligned}$$

where $0 < \alpha_3 < 1$. So we have

$$\hat{U}(\hat{\gamma}[t+1]) - \hat{U}(\hat{\gamma}[t]) \geq ||\nabla\mathcal{L}_{\lambda[t]}(\hat{\gamma}[t])||^2\{h[t] - 0.5[V_\lambda V_\rho + V_U(L+1)]h[t]^2\}.$$

□

Bibliography

[1] V. D. Blondel, L. Ninove, and P. Van Dooren. An affine eigenvalue problem on the nonnegative orthant. Linear Algebra and its Applications, 404:69-84, 2005..

[2] U. Krause. Concave Perron-Frobenius theory and applications. Nonlinear analysis, 47(2001):1457-1466, 2001.

[3] U. Krause. Perron's stability theorem for non-linear mappings, J. Math. Econ. 15(1986), 275-282.

[4] C. W. Tan, Nonconvex Power Control in Multiuser Communication Systems, Ph.D. Thesis, Nov. 2008, Princeton University, Electrical Engineering Dept.

[5] C. W. Tan, M. Chiang and R. Srikant, Fast Algorithms and Performance Bounds for Sum Rate Maximization in Wireless Networks, Proc. IEEE INFOCOM, Rio de Janerio, Brazil, Apr 20-25, 2009.

[6] C. W. Tan, S. Friedland and S. H. Low, Spectrum Management in Multiuser Cognitive Wireless Networks: Optimality and Algorithm, IEEE Journal on Selected Areas in Communications, Vol. 29, No. 2, pp. 421-430, Feb. 2011.

[7] C. W. Tan, M. Chiang and R. Srikant, Maximizing Sum Rate and Minimizing MSE on Multiuser Downlink: Optimality, Fast Algorithms, and Equivalence via Max-min SINR, IEEE Transactions on Signal Processing, Vol. 59, No. 12, pp. 6127-6143, Dec. 2011.

[8] C. W. Tan, S. Friedland and S. H. Low, Nonnegative Matrix Inequalities and Their Application to Nonconvex Power Control Optimization, SIAM Journal on Matrix Analysis and Applications, accepted for publication.

[9] Y. Xiao and F. Hu, Cognitive Radio Networks. CRC Press, 2009.

[10] C. D. Meyer, Matrix Analysis and Applied Linear Algebra. Society for Industrial and Applied Mathematics, 2000.

[11] S. Friedland. Convex spectral functions. Linear and Multilinear Algebra, 9(4):299¨C316, 1981.

[12] M. Grant and S. Boyd, cvx Users' Guide for cvx version 1.22, August 2012.

[13] J. Zhang, Matlab m file "logPFeig.m", attached.

[14] S. Boyd, Subgradients, lecture notes of EE364b, Standford University, available at http://www.stanford.edu/class/ee364b/lectures/subgradients_slides.pdf.

[15] S. Friedland and S. Karlin. Some inequalities for the spectral radius of non-negative matrices and applications. Duke Mathematical Journal, 42(3):459–490, 1975.

[16] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. IEEE Trans. on Information Theory, 52(6):2508–2530, 2006.

[17] B. Song, R. L. Cruz and B. D. Rao, Network Duality for Multiuser MIMO Beamforming Networks and Applications, IEEE Trans. on Communications, vol. 55, NO. 3, Mar. 2007.

[18] http://en.wikipedia.org/wiki/Reciprocity_(electromagnetism)

[19] S. Boyd and L. Vandenberghe, Convex Optimization, Cambridge, U. K.: Cambridge Univ. Press, 2004.

[20] M. Chiang, P. Hande, T. Lan and C. W. Tan, Power Control in Wireless Cellular Networks, Foundations and Trends in Networking, 2(4):381-533, 2008.

[21] G. J. Foschini and Z. Miljanic, A Simple Distributed Autonomous Power Control Algorithm and its Convergence, IEEE Trans. on Vehicular Technology, 42(4):641-646, 1993.

[22] Y. Nesterov, Introductory Lectures on Convex Optimization. A Basic Course, Kluwer Academic Publishers, 2004.

[23] D. P. Bertsekas, Nonlinear Programming, 2nd ed. Belmont, MA: Athena Scientific, 1999.

[24] D. P. Bertsekas, Convex Optimization Theory, Belmont, MA: Athena Scientific, 2009. Its supplementary chapter 6 is available online at http://www.athenasc.com/convexdualitychapter.pdf.

[25] P. Hande, S. Rangan, M. Chiang and X. Wu, Distributed Uplink Power Control for Optimal SIR Assignment in Cellular Data Networks, IEEE/ACM Transactions on Networking, Vol. 16, No. 6, Dec. 2008.

[26] 3Gpp TR 36.814 V9.0.0 (2010-03), Evolved Universal Terrestrial Radio Access (E-UTRA); Further advancements for E-UTRA Physical layer aspects.

[27] M. Chiang, C. W. Tan, D. P. Palomar, D. O'Neill and D. Julian, Power Control by Geometric Programming, IEEE Transactions on Wireless Communications, Vol. 6, No. 7, pp. 2640-2651, Jul. 2007.

[28] D. W. H. Cai, T. Quek and C. W. Tan, Coordinated Max-Min SIR Optimization in Multicell Downlink - Duality and Algorithm, IEEE ICC 2011.

[29] C. W. Tan, Optimal Power Control in Rayleigh-fading Heterogeneous Networks, IEEE INFOCOM 2011.

[30] R. D. Yates, A Framework for Uplink Power Control and Active Link Quality Protection, IEEE J. Se. Areas Commun.,vol 13, no. 7, pp. 1341-1347, Sep. 1995.

[31] J. Huang, R. Berry and M. L. Honig, "Distributed Interference Compensation for Wireless Networks", IEEE Journal on Selected Areas in Communications. 24 (2006): 1074-1084.

[32] H. Boche, S. Naik and T. Alpcan, “Characterization of convex and concave resource allocation problems in interference coupled wireless systems”, IEEE Transactions on Signal Processing. 59 (5): 2382-2394.